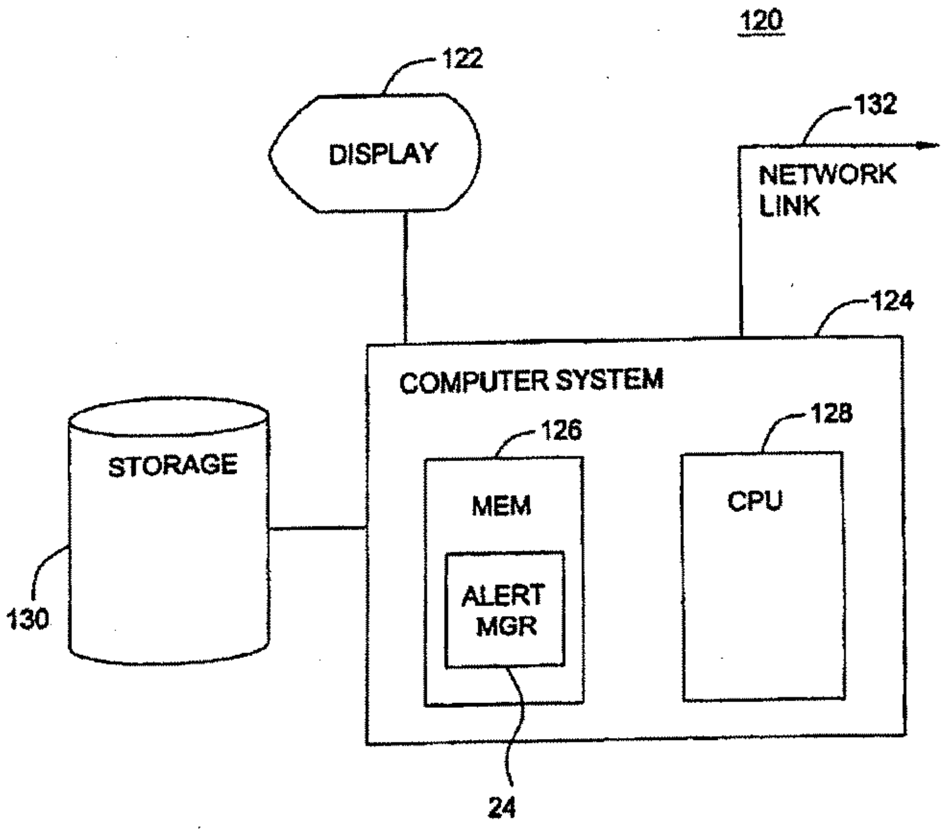

120
122
DISPLAY
132
NETWORK LINK
124
COMPUTER SYSTEM
126
128
STORAGE
MEM
CPU
ALERT MGR
130
24

FIG. 6

# NETWORK-BASED ALERT MANAGEMENT

This application claims priority under 35 USC §120 to U.S. patent application Ser. No. 09/188,739, filed on Nov. 9, 1998, now U.S. Pat. No. 6,321,338, the entire contents of which are hereby incorporated by reference.

## GOVERNMENT RIGHTS IN THIS INVENTION

This invention was made with U.S. government support under contract numbers F30601-96-C-0294 and F30602-99-C-0187 awarded by the U.S. Air Force Research Laboratory. The U.S. government has certain rights in this invention.

## TECHNICAL FIELD

This invention relates to network-based alert management.

## BACKGROUND

Computer networks may include one or more digital security monitors or sensors that automatically analyze traffic on the network to identify potentially suspicious activity. The sensors may be implemented in either software or hardware. Monitors may focus on security monitoring and/or on fault analysis.

Upon detecting suspicious activity, the sensors typically generate some kind of digital alert message or signal, and attempt to bring that message to the attention of network I/S managers whose responsibility it is to respond and react in an appropriate defensive manner against hostile digital attacks or to recover quickly from catastrophic failures.

## SUMMARY

In an aspect, the invention features a method of managing alerts in a network including receiving alerts from network sensors, consolidating the alerts that are indicative of a common incident and generating output reflecting the consolidated alerts. Alerts are formatted into a standard alert format by the network sensors or an input receiving logic of an alert management system, or a combination of both. The alert format may be selected from a group of formats including IDIP, SNMP, HP OpenView, Attach Specification CIDF and GfDO. Alerts may be tagged with corresponding significance scores where the significance scores may include a priority measure for the corresponding alerts. The priority measure may be derived from a priority map that can be automatically generated or dynamically adjusted. The priority map may contain relative priority scores for resource availability, resource integrity and resource confidentiality.

In another aspect, the invention features a method of managing alerts including receiving alerts from a number of network sensors, filtering the alerts to produce one or more internal reports and consolidating the internal reports that are indicative of a common incident-to-incident report. Related incident reports may be correlated. The network sensors may format the received alerts. Filtering includes deleting alerts that do not match specified rules. The filtering rules may be dynamically adjusted. Filtering may also include tagging alerts with a significance score that can indicate a priority measure and relevance measure.

Among the advantages of the invention may be one or more of the following.

The alert manager can be tailored to a particular application by dynamically adding or removing data connections to sources of incoming alerts, and by dynamically varying the

process modules, user filter clauses, priority clauses, topology clauses, and output. Process modules may be added, modified, and deleted while the alert manager is active. Output may be configured for a variety of graphical user interfaces (GUIs). In embodiments, useful, for example, for each category of attack the user can define different priorities as related to denial of service, security, and integrity.

Process modules are logical entities within the alert manager that can respond to an incoming alert in real time and virtual time, i.e., data within an application can cause the alert manager to respond.

The alert manager can act as a sender or receiver. In embodiments, useful, for example, the alert manager can listen to a specific port in a network or connect to an external process on a host computer and process its data.

The alert management process can be an interpretive process allowing the incorporation of new process clauses and new rules.

The alert management process may provide a full solution for managing a diverse suite of multiparty security and fault monitoring services. Example targets of the alert management process are heterogeneous network computing environments that are subject to some perceived operational requirements for confidentiality, integrity, or availability. Inserted within the network are a suite of potential multiparty security and fault monitoring services such as intrusion detection systems, firewalls, security scanners, virus protection software, network management probes, load balancers, or network service appliances. The alert management process provides alert distributions within the monitored network through which security alerts, fault reports, and performance logs may be collected, processed and distributed to remote processing stations (e.g., Security Data Centers, Administrative Help Desks, MIS stations). Combined data produced by the security, fault, or performance monitoring services provide these remote processing stations detailed insight into the security posture, and more broadly the overall health, of the monitored network.

Value may be added to the content delivered by the alert management process to the remote processing station(s) that subscribe to alerts in the form of an advanced alert processing chain. For example, alerts received by the alert management process and prepared for forwarding to a remote processing station, may be filtered using a dynamically downloadable message criteria specification.

In a further aspect, alerts may be tagged with a priority indication flag formulated against the remote processing station's alert processing policy and tagged with a relevance flag that indicates the likely severity of the attack with respect to the known internal topology of the monitored network.

In a further aspect of the invention, alerts may be aggregated (or consolidated) into single incident reports when found to be associated with a series of equivalent alerts produced by the same sensor or by other sensors, based upon equivalence criteria, and the incident reports forwarded to the remote processing station.

The alert management system is configurable with respect to the data needs and policies specified by the remote processing station. These processes are customizable on a per remote processing station basis. For example, two remote processing stations may in parallel subscribe to alerts from the alert management process, with each having individual filtering policies, prioritization schemes, and so forth, applied to the alert/incident reports it receives.

Other features and advantages will become apparent from the following description and from the claims.

## DESCRIPTION OF DRAWINGS

FIG. 1 is a block diagram of a network based alert management system.

FIG. 2 is a flow diagram of an alert management process.

FIG. 3 is a block diagram of a priority database record.

FIG. 4 is a block diagram of the remote processing center.

FIG. 5 is a block diagram of a software architecture for the alert management system.

FIG. 6 is a block diagram of a computer platform.

Like reference symbols in the various drawings indicate like elements.

## DETAILED DESCRIPTION

Referring to FIG. 1, a network based alert management system 10 includes a network 12, a network 14, and a network 16. Each of the networks 12–14 includes a number of computer systems collectively labeled 18, interconnected, for example, by an Ethernet cable 20. Each of the networks 12–14 includes security and fault monitoring systems generally labeled 22. Each security and fault monitoring system 22 is linked to an alert manager 24. The alert manager 24 is linked to one or more remote processing centers generally labeled 26. Each alert processing center 26 includes a remote management interface 36 (shown on only one center 26 by way of example). The remote management interface 36 provides a user (not shown) the capability of configuring reports produced by the alert manager 24.

The security and fault monitoring systems 22 may include, for example, intrusion detection systems, firewalls, security scanners, virus protection software, network management probes, load balancers, and network service appliances. Each of the security and fault monitoring systems 22 produces an alert stream in the form of, for example, security alerts, fault reports, and performance logs. The alert stream is sent to the alert manager 24 for collection, processing, and distribution to the remote processing center 26. Example remote processing centers 26 are security data centers, administrative help desks, and MIS stations.

In an embodiment, the remote processing center 26 subscribes to the alert manager 24 which in turns distributes specific collected and processed alert information to the remote processing center 26, more fully described below.

The networks 14, 14, and 16 being monitored by the security and fault monitoring systems 22 may include any computer network environment and topology such as local area networks (LAN), wide area networks (WAN), Ethernet, switched, and TCP/IP-based network environments. Network services occurring within the networks 12–16 include features common to many network operating systems such as mail, HTTP, ftp, remote log in, network file systems, finger, Kerbebos, and SNMP. Each of the sensors 22 monitors various host and/or network activity within the networks 12–16, and each sensor 22, as discussed above, generate a stream of alerts, triggered by potentially suspicious events, such as network packet data transfer commands, data transfer errors, network packet data transfer volume, and so forth. The alerts indicate a suspicion of possible malicious intrusion or other threat to operations within the networks 12–16.

The alert manager 24 includes a receive-input logic module 28. In an embodiment, the receive-input logic 28 of the alert manager 24 subscribes, i.e., establishes a transport connection, to receive each of the alert streams produced by the sensors 22 through a secure electronic communication line (SSL) 30. The alert streams contain raw, i.e., unprocessed, alerts. The monitors 22 may format their respective alert streams in a variety of formats, such as IDIP, SNMP, HP Openview, an XML-based standard format (such as the Attack Specifications from IETF), Common Intrusion Detection Framework (CIDF), GIDOs, or some other format. The receive-input logic 28 of the alert manager 24 is equipped with translation modules 32 to translate the original, raw alert streams from the monitors 22 into a common format for further processing, if the alerts do not arrive in the common format.

In another embodiment, the monitors 22 include conversion software (not shown), also referred to as "wrapper" software that translates a monitor's raw alert stream into the common format used by the alert manager 24. The wrapper software can add data items of interest to the alert manager 24, by querying its network 12–16.

In another embodiment, a combination of monitors 22 having wrapper software and the receive-input logic 28 preprocessing raw alerts in the alert management network 10 are present to accommodate a heterogeneous base of monitors 22 that an end-user desires to manage.

The alert manager 24 includes an alert processing engine 34. Raw alerts received by the receive-input module 28 and formatted into the common format are sent to the alert processing engine 34.

Referring to FIG. 2, an alert management process 50 residing in the alert processing engine 34 includes receiving 52 formatted alerts from the receive-input logic 28. The formatted alerts are passed 54 through user-specified filters and alerts not matching criteria of the user-specified filters are discarded.

For example, a particular end-user subscriber may be responsible only for a portion of the overall operations network and may only wish to see alerts coming from a particular subset of monitors 22, e.g., from particular ports. Each end-user subscriber can interactively define his or her own customized user-specified filters using the remote management interface 36 of the remote processing center 26, fully described below.

The filtered alerts are prioritized 56, i.e., rated or scored according to priorities dynamically controlled by the user. In an embodiment, the priority of an alert is determined by analyzing the known, (relative) potential impact of the attack category identified with respect to each of various concerns such as confidentiality, data integrity, and system availability. Confidentiality involves allowing only authorized users to view network data. Data integrity involves allowing only authorized persons to change data. System availability involves providing users access to data whenever needed with minimum downtime.

Different categories of known computer intrusions and anomalies generally pose threats with differing levels of impact on each of the above three concerns. In addition, for different users and different applications, each of the concerns may be of different relative priority. For example, in a general Internet news/search portal like Yahoo! or Lycos, continuous availability may be a more important concern than confidentiality. Conversely, for a government intelligence database, confidentiality may be a greater priority than continuous availability. For an e-commerce business site, all three concerns may be of roughly equal seriousness and priority. An ultimate priority score assigned to a particular alert for a given end-user during prioritization 56 reflects a sum or combination of the identified attack's potential adverse impact along each of the dimensions of interest (confidentiality, data integrity, and system availability), weighted by the end-user's individual profile of relative priority for each such dimension.

In an embodiment, a default priority profile is provided for each user or subscriber that assigns equal priority to confidentiality, data integrity, and system availability. In a preferred embodiment, the end-user may configure the priorities dynamically, and modify the default values as desired, through the remote management interface 36 that gives the user the flexibility to customize priority assignments in a manner that reflects his/her unique concerns.

In an another embodiment, users (or system developers) directly assign a relative priority score to each type of attack, instead of ranking more abstract properties such as integrity or availability, that allows more precise reflection of a user's priorities regarding specific attacks, but requires more initial entry of detailed information.

In an embodiment, users may register a listing of critical services, identified by <host ID, protocol> pairs, as to whom potential attacks or operational failures are considered to be of especially high priority.

Management and alteration of filters and listings of critical services in accordance with each of the prioritization methodologies described above can are performed dynamically and interactively while alert manager 24 is in operation and as user priorities change using the remote management interface 36.

The alerts are topology vetted 58. Vetting 58 provides a relevance rating to alerts based on the topological vulnerability of the network being monitored to the type of attack signaled by the alert. Example topologies include the computing environment, what kind of operating system (O/S), network infrastructure, and so forth. In a preferred embodiment, vetting 58 utilizes a mapping between each network host and an enumeration of that host's O/S and O/S version(s). Vetting step 58 further preferably utilizes a topology relevance table indicating the relevance of various types of attacks to each of the different possible OS/version environments. Thus, to determine and assign a relevance score for a particular alert, the host ID (hostname/IP address) for the target of that alert can be used to retrieve its OS/version information, and the OS/version along with the attack type of the alert can be used to retrieve a relevancy score from the topology table.

In an embodiment, the topology table of the network being monitored is dynamically configurable by end users through the remote management interface 36.

In another embodiment, automatic local area network (LAN) mapping is provided by a network topology scope application.

The relevance of various types of known attacks against different topologies is preferably specified in predefined maps, but dynamically configured using the remote management interface 36.

Internal reports are generated 60 from the output of filtering 54, prioritizing 56 and vetting 58. Internal reports generally include fewer alerts as compared with the original raw alert stream, as a result of the user-configured filtering 40. Internal reports also tag or associate each alert with priority and/or relevance scores as a result of priority mapping 56 and topology vetting 58, respectively.

The internal reports are used to generate 62 consolidated incident reports. A consolidated incident report adds perspective and reduces information clutter by merging/combining the internal reports for multiple alerts into a single incident report. In a preferred embodiment, generat-

ing 62 is carried out through report aggregation and equivalence recognition. Aggregation refers to combining alerts produced by a single sensor, whereas equivalence recognition refers to combining alerts from multiple sensors.

The underlying notion in both cases is that nominally different alerts may actually represent a single intrusion "incident" in the real world. By analogy, a single criminal intrusion into a physical property might trigger alarms on multiple sensors such as a door alarm and a motion detector that are instrumented on the same premises, but from an informational perspective both alarms are essentially signaling the same event.

In an embodiment, alert parameters examined for report aggregation include a variable combination of attack type, timestamp, monitor identification (ID), user ID, process ID, and <IP, port addresses> for the source and target of the suspicious activity.

When an internal report is generated 60 alerts are consolidated and the corresponding priority and relevance tags for the individual alerts are merged into single meta-priority/meta-relevance scores for the single incident. Different functions may be utilized for doing the priority blend, such as additive, min/max, average, and so forth. Duration of the overall incident is also preferably computed and associated with the incident, based on the time stamps of the various individual alerts involving the incident.

The consolidated incident reports are used to generate 64 a report output. Formatting of the output report is based on subscriber-customized criteria that are defined using the remote management interface 36. The report output is transported 66 to the remote processing center 26.

Selection of a transport is under user control and managed using the remote management interface 36. The user may specify, for example, E-mail, XML, HTML and/or writing out to a file. In an embodiment, the transport occurs over an SSL for display and assessment by the end-user.

The filtering 54, prioritization 54 and topology vetting 58 are event driven, i.e., each alert is processed and filtered/tagged as it arrives, one alert at a time. However, temporal clauses are utilized for aspects of report aggregation and equivalence recognition among multiple alerts. For example, as internal reports are generated 60 a sliding window is established during which additional records may be merged into the aggregate incident report. A single-alert internal report may be sent to the remote processing center 26 indicating that it has witnessed the alert. A subsequent aggregate alert report, i.e., an incident report, covering that single alert as well as others, may also be forwarded to the remote processing center 26 to indicate a duration of the attack/incident, an aggregate count of individual alerts representing this incident, and an aggregate priority. In an embodiment, aggregate alert flushing may occur after some period of inactivity (e.g., "two minutes since last event"). The aggregate alert flushing is not event driven, but rather driven by an internal timeout recognized from a system clock (not shown) of the alert manager 24.

Referring to FIG. 3, an exemplary priority database record 80 used for prioritization 56 of filtered alerts includes example network attacks ping of death 82, buffer overflow 84 and write polling violation 86. For each of the attacks 82–86, a relative priority rating is assigned, namely, denial of service (system availability) 88, data integrity 90, and security (confidentiality) 92. By way of example, a first end-user 94 weights denial of service at 0%, data integrity at 20%, and security at 80%. A second end-user 96 weights denial of service at 80%, data integrity at 10% and security at 10%. Thus, for the priority database record 80, the user 94 emphasizes a high concern (priority) with security, while the user 96 emphasizes a high concern (priority) with denial of service.

In this example, for first user 94 a "ping of death" alert 82 will have a priority score=(0*90)+(0.2*10)+(0.8*10)=10; whereas for second user 96 a "ping of death" alert 82 will receive a priority score=(0.8*90)+(0.1*10)+(0.1*10)=74.

As is seen from the description above, (a) it is the relative value of these priority scores that has significance, not the absolute magnitudes, and (b) the priority values for alerts and for user preferences are subjective values that may vary from one application to another and from one user to another. As noted above, the alert priority map values and user priority profiles may be dynamically adjusted and customized by individual users via remote management interface 36.

Referring again to FIG. 1, the report output of the alert processing process 50 is stored at the remote processing center 26 in a database 38 contained in a storage device 40 for retrieval and reporting by the end user. In an embodiment, the report output is translated at the remote processing center 26 in accordance with a user-configurable database schema into an existing/legacy database management system (not shown) contained in the remote processing center 26 for convenience of the end-user, either manually by a database integration team or automatically using a database mediator/translator. The remote management interface 36 accesses the database management system and presents the report output to the end-user, such as by a graphical user interface (GUI) on a workstation 42.

In an embodiment, the alert management network 10 provides an open, dynamic infrastructure for alert processing and management. The alert manager 24 preferably includes functionality for dynamically generating, suspending, and configuring data connections and logical process modules, in response to interactive remote user commands issued via remote management interface 36. The remote management interface 36 preferably executes a java application that generates command files, in response to end user requests, in the form of directives and any necessary data files, such as the priority database record 80, and so forth. The java application communicates, e.g. via telnet, to the alert manager 24 and downloads the directives and data files. The alert processing engine 34, preferably a postscript interpreter in one embodiment, can process the directives dynamically. Many of the directives are preferably defined in terms of postscript code that resides locally in a library 44 in the alert manager 24. Applications running in alert manager 24 are written in modular fashion, allowing directives to accomplish meaningful change of logical behavior by instructing the alert manager 24 to terminate a particular process clause and activate a newly downloaded clause, for example.

By way of another example, through the operation of the alert processing engine 34 the alert manager 24 can dynamically establish and suspend connections to the various alert streams generated by the security and fault monitoring systems 22. Thus, the alert manager 24 can dynamically "plug into" (i.e., connect) new alert streams, such as alert streams from additional sensors newly deployed by an end-user, and likewise can dynamically suspend (permanently or temporarily) its connection to alert streams from sensors 22 that are removed, replaced, taken offline, and so forth. Similarly, alert manager 24 can dynamically generate or suspend modules of the alert management process 50, and can dynamically adjust the configurable parameter settings of those modules.

In this manner, alert manager 24 is designed to be responsive to dynamic configuration requests initiated by end users using the remote management interface 36 of the remote processing center 26. As mentioned above, the remote management interface 36 provides an interactive interface at workstation 42 for end-users to specify desired modifications to the dynamically configurable aspects of alert manager 24.

Referring to FIG. 4, a block diagram of a software architecture 100 for a dynamic, open, alert management infrastructure in accordance with preferred embodiments of the present invention is shown. An infrastructure module 102 (labeled "eFlowgen") provides core infrastructure functionality, including implementation of the alert processing engine 34, and need not be specialized to alert management applications. An inline application code module 104 (in conjunction with an initialization module 106, described below) defines an alert management application, including the overall alert analysis and reporting process 50 described above with reference to FIG. 2. Initialization script module 106 complements application code module 104, by defining, for a particular application instance, the specifics of the input/output transport connections and specifics of the logical alert processing clauses corresponding to the process 50. A dynamic definitions module 108 represents dynamic changes submitted by users via the remote management interface 36, such as configuration changes and other extensions as previously discussed; the functionally dynamic definitions module 180 are comparable to initialization script module 106, except for being dynamically submitted and incorporated into the running application.

A detailed functional specification for a software infrastructure corresponding to eFlowgen module 102 is included in the Appendix, incorporated herein.

In another embodiment, referring to FIG. 5, the remote processing center 26 includes a correlation logic engine 110. The correlation logic engine 110 accesses and compares incident reports in database 38 and attempts to provide intelligent assistance to end-users in the analytical task of discovering patterns and making sense of alert data. The correlation engine logic 110 looks for key attribute relations in common for different incidents, such as incidents targeting a single host machine over a relatively short time frame, or incidents reflecting attacks or anomalies coming from a particular source machine. Automatically correlating separate incidents helps end-users recognize more quickly that a particular machine is under serious attack or that some other machine is a hostile "bad guy," for example, and the end-users can then take appropriate defensive action.

Another correlation technique residing in the correlation logic engine 110 looks for interrelated vulnerabilities, applying rule-based knowledge to look for groups of distinct incidents that can inferentially be interpreted as related parts of a single, coordinated attack. For example, rules matching patterns of incidents that look like a chain over time, where the target of an earlier incident becomes the source of a subsequent incident, may allow correlation logic engine 110 to conclude that these likely are not unrelated incidents, and that a "worm" infection appears to be spreading.

In an embodiment, the correlation logic engine 110 incorporates statistical inferential methods. The correlation logic engine 110 attempts to draw conclusions automatically based on received intrusion incident reports. The correlation logic engine 110 produces reports for the end-user indicating correlation found.

The alert manager 24 and other components of the alert management network 10 may be implemented and executed on a wide variety of digital computing platforms, including, but not limited to, workstation-class computer hardware and operating system software platforms such as Linux, Solaris, FreeBSD/Unix, and Windows-NT.

Referring to FIG. 6, a computer platform 120 suitable for hosting and executing the alert management process 50 includes a display device 122 connected to a computer 124. The computer 124 includes at least a memory 126 and a central processing unit (CPU) 128. The computer 124 includes a link to a storage device 130 and a network link 132.

The storage device 130 can store instructions that form an alert manager 24. The instructions may be transferred to the memory 126 and CPU 128 in the course of operation. The instructions for alert manager 24 can cause the display device 122 to display messages through an interface such as a graphical user interface (GUI). Further, instructions may be stored on a variety of mass storage devices (not shown).

Other embodiments are within the scope of the following claims.

What is claimed is:

1. A computer-implemented method of managing alerts in a network comprising:

receiving alerts from network sensors;

consolidating the alerts that are indicative of a common incident; and

generating output reflecting the consolidated alerts.

2. The computer-implemented method of claim 1 further comprising formatting the alerts into a standard alert format.

3. The computer-implemented method of claim 2 wherein formatting the alerts into a standard alert format is performed by the network sensors.

4. The computer-implemented method of claim 2 wherein formatting the alerts into a standard alert format is performed by input-receiving logic of an alert management system.

5. The computer-implemented method of claim 2 wherein the alert format is selected from the following group of formats: {IDIP, SNMP, HP Openview, Attach Specification, CIDF, XML}.

6. The computer-implemented method of claim 1 further comprising tagging the alerts with corresponding significance scores.

7. The computer-implemented method of claim 6 wherein the significance scores comprise a component indicating a priority measure for the corresponding alerts.

8. The computer-implemented method of claim 7 wherein the priority measure is derived using a priority map.

9. The computer-implemented method of claim 8 wherein the priority map is dynamically adjustable.

10. The computer-implemented method of claim 8 wherein the priority map comprises relative priority scores for resource availability, resource integrity, and resource confidentiality.

11. The computer-implemented method of claim 7 wherein the priority measure is derived based upon criticality of one or more resources targeted by the corresponding alerts.

12. The computer-implemented method of claim 6 wherein the significance scores comprise a component indicating a relevance measure for the corresponding alerts.

13. The computer-implemented method of claim 12 wherein the relevance measure is derived based upon a consideration of an operating environment topology for a target of an attack signaled by the corresponding alert.

14. The computer-implemented method of claim 13 wherein the relevance measure is derived using one or more topology tables.

15. The computer-implemented method of claim 14 wherein one or more-elements of the topology tables are dynamically adjustable.

16. The computer-implemented method of claim 14 wherein one or more elements of the topology tables are automatically generated.

17. The computer-implemented method of claim 14 wherein the one or more topology tables comprise:

a mapping between one or more network hosts and one or more corresponding environment features selected from the following group: {operating systems (o/s), o/s versions, hosted services/applications}; and

a relevance rating for each of one or more types of attacks mapped against the corresponding environment features.

18. The computer-implemented method of claim 6 wherein the output reflecting the consolidated alerts includes a meta-significance score reflecting a blending of the significance scores for each of the consolidated alerts.

19. The computer-implemented method of claim 1 further comprising correlating common incidents.

20. The computer-implemented method of claim 1 further comprising filtering the alerts.

21. The computer-implemented method of claim 20 wherein filtering comprises comparing the alerts to user-specified filters.

22. The computer-implemented method of claim 21 wherein the user-specified filters are dynamically configurable.

23. The computer-implemented method of claim 1 wherein the consolidated alerts comprise alerts produced by a single network sensor.

24. The computer-implemented method of claim 1 wherein the consolidated alerts comprise alerts produced by different network sensors.

25. The computer-implemented method of claim 1 wherein consolidating the alerts further comprises identifying the alerts that are indicative of a common incident based upon one or more alert parameters selected from the following group: {attack type, timestamp, network security component identification (ID), user ID, process ID, <IP, port addresses> for a source and a target of a suspicious activity}.

26. The computer-implemented method of claim 1 wherein generating output comprises generating one or more subscriber-specific reports.

27. The computer-implemented method of claim 26 wherein the subscriber-specific reports are based on one or more subscriber-customizable criteria.

28. The computer-implemented method of claim 27 wherein the subscriber-customizable criteria are dynamically configurable.

29. The computer-implemented method of claim 27 wherein the subscriber-customizable criteria comprise one or more transport options.

30. The computer-implemented method of claim 29 wherein the transport options comprise one or more options selected from the following group: {E-mail, XML, HTML, writing out to a file}.

31. The computer-implemented method of claim 1 wherein the output is automatically input to a data base management system.

32. The computer-implemented method of claim 1 further comprising sending the output over one or more secure communications links to one or more subscribers.

33. The computer-implemented method of claim 1 wherein receiving alerts further comprises dynamically modifying a set of network sensors from whom the alerts are received.

34. The computer-implemented method of claim 1 wherein the network sensors comprise heterogeneous network sensors.

35. The computer-implemented method of claim 1, wherein the received alerts include one or more filtered alerts.

36. The computer-implemented method of claim 1, wherein the received alerts include one or more alerts tagged with corresponding significance scores.

37. The computer-implemented method of claim 1, wherein the received alerts include one or more consolidated alerts, as to which the method of claim 1 is applied in recursive fashion.

38. The computer-implemented method of claim 1, further comprising processing the alerts to produce one or more internal reports, and wherein consolidating comprises consolidating the internal reports that are indicative of a common incident to produce one or more incident reports.

39. A computer program, residing on a computer-readable medium, comprising instructions causing a computer to:

receive alerts from a plurality of network sensors;

consolidate the alerts that are indicative of a common incident; and

generate output reflecting the consolidated alerts.

40. The computer program of claim 39, further comprising instructions causing a computer to:

format the alerts;

filter the alerts; and

tag one or more of the alerts with corresponding significance scores.

41. The computer program of claim 39 wherein the network sensors comprise heterogeneous network sensors.

42. In a computer network that has a plurality of security or fault monitoring devices of various types, each of which generates an alert when an attack or anomalous incident is detected, a method for managing alerts comprising the steps of:

ranking network resources and services based on their actual or perceived importance to effective operation of the network;

receiving alerts from the security or fault monitoring devices;

ranking each alert based on a potential or actual impact of each alert's underlying attack or incident on effective operation of the network;

grouping alerts that may relate to a common attack or incident; and

generating a report that shows at least a subset of the alert groups and that indicates a potential or actual impact of each alert group's underlying attack or incident on effective operation of the network.

43. In a computer network that has a plurality of security or fault monitoring devices of various types, each of which generates an alert when an attack or anomalous incident is detected, a method for managing alerts comprising the steps of:

ranking network resources and services based on their actual or perceived importance to effective operation of the network;

receiving alerts from the security or fault monitoring devices;

grouping alerts that may relate to a common attack or incident;

ranking each alert group based on a potential or actual impact of each alert group's underlying attack or incident on effective operation of the network; and

generating a report that shows at least a subset of the alert groups and that indicates a potential or actual impact of each alert group's underlying attack or incident on effective operation of the network.

44. In a computer network that has a plurality of security or fault monitoring devices of various types, each of which generates an alert when an attack or anomalous incident is detected, a method for managing alerts comprising the steps of:

receiving alerts from the security or fault monitoring devices;

grouping alerts that may relate to a common attack or incident;

ranking each alert group based on a potential or actual impact of each alert group's underlying attack or incident on effective operation of the network; and

generating a report that shows at least a subset of the alert groups and that indicates a potential or actual impact of each alert group's underlying attack or incident on effective operation of the network.

45. The method of claim 44 wherein the security or fault monitoring devices are selected from the following group of devices:

firewalls;

intrusion detection systems;

antivirus software;

security scanners;

network management probes;

network service appliances;

authentication services; and

host and application security services.

46. The method of claim 44 further comprising the step of identifying critical network services and resources.

47. The method of claim 44 further comprising the step of ranking network resources and services based on their actual or perceived importance to effective operation of the network.

48. The method of claim 44 further comprising the step of identifying a set of alert classes or types.

49. The method of claim 48 wherein the set of alert classes or types is selected from the following group:

privilege subversion;

use subversion;

denial of service;

intelligence gathering;

access violations;

integrity violations;

system environment corruption;

user environment corruption;

asset distress; and

suspicious usage.

50. The method of claim 49 further comprising the step of ranking the alert classes or types based on actual or perceived impact of the underlying attacks or incidents on effective operation of the network.

51. The method of claim 44 wherein the alerts are grouped based on alert attributes selected from the following group:

common source;

common connection;

common host-based session;

common alert type or class; and

information about alert equivalence from an external data base.

52. The method of claim 44 wherein the alert groups are ranked based on criteria selected from the following group:

attack outcome;

attack vulnerability;

target of the attack;

alert class;

attacker identity; and

user identity.

53. The method of claim 52 wherein the criteria are assigned weights that are dynamically adjustable.

54. The method of claim 44 wherein the alert report further includes information selected from the following group:

alert class;

alert group rank,

duration of the attack or incident; and

name, location, and version of the security or fault monitoring devices that generated alerts.

55. In a computer network, a method for ranking alerts that are indicative of an attack or an anomalous incident, the method comprising the steps of:

identifying and ranking different types of attacks or incidents according to their actual or perceived impact on effective operation of the network;

identifying and ranking network resources or services according to their actual or perceived importance to effective operation of the network;

determining vulnerability of network resources to different types of attacks or incidents; and

assigning a relevance score to an alert based on the type of the underlying attack or incident, the target of the attack-or incident, and the vulnerability of the target.

56. The method of claim 55 wherein an attack outcome measurement is also used in the assignment of the relevance score.

57. The method of claim 55 wherein the relevance score is assigned in part by associating dynamically adjustable weights with the type of the underlying attack or incident, the target of the attack or incident, and the vulnerability of the target.

* * * * *

(12) **United States Patent**
**Porras et al.**

(10) **Patent No.: US 6,711,615 B2**
(45) **Date of Patent: *Mar. 23, 2004**

(54) **NETWORK SURVEILLANCE**

(75) Inventors: **Phillip Andrew Porras**, Moutain View, CA (US); **Alfonso Valdes**, San Carlos, CA (US)

(73) Assignee: **SRI International**, Menlo Park, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **10/254,457**

(22) Filed: **Sep. 25, 2002**

(65) **Prior Publication Data**

US 2003/0088791 A1 May 8, 2003

**Related U.S. Application Data**

(63) Continuation of application No. 09/658,137, filed on Sep. 8, 2000, now Pat. No. 6,484,203, which is a continuation of application No. 09/188,739, filed on Nov. 9, 1998, now Pat. No. 6,321,338.

(51) **Int. Cl.**$^7$ .......................... **G06F 11/30**; G06F 12/14
(52) **U.S. Cl.** .......................................... **709/224**; 713/201
(58) **Field of Search** ................................ 713/200, 201; 709/223–225

(56) **References Cited**

U.S. PATENT DOCUMENTS

| | | | | |
|---|---|---|---|---|
| 4,672,609 | A | 6/1987 | Humphrey et al. | 371/21 |
| 4,773,028 | A | 9/1988 | Tallman | 364/550 |
| 5,210,704 | A | 5/1993 | Husseiny | 364/551.01 |
| 5,440,723 | A | 8/1995 | Arnold et al. | 395/181 |
| 5,539,659 | A | 7/1996 | McKee et al. | 709/224 |
| 5,557,742 | A | 9/1996 | Smaha et al. | 395/186 |
| 5,706,210 | A | 1/1998 | Kumano et al. | 709/224 |
| 5,748,098 | A | 5/1998 | Grace | 340/825.16 |

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

| | | | |
|---|---|---|---|
| WO | 99/13427 | 3/1999 | G06K/7/00 |
| WO | 99/57626 | 11/1999 | G06F/1/16 |
| WO | 00/10278 | 2/2000 | |
| WO | 00/25214 | 5/2000 | G06F/12/14 |
| WO | 00/25527 | 5/2000 | H04Q/3/00 |
| WO | 00/34867 | 6/2000 | G06F/11/30 |
| WO | 02/101516 | 12/2002 | |

OTHER PUBLICATIONS

Debar, et al., "Towards a Taxonomy of Intrusion–Detection Systems," Computer Networks 31 (1999), 805–822.
Debar et al., "A Neural Network Component for an Intrusion Detection System," © 1992 IEEE.
Denning et al, "Prototype IDES: A Real–Time Intrusion–Detection Expert System," SRI Project ECU 7508, SRI International, Menlo Park, California, Aug. 1987.
Denning et al., "Requirements and Model for IDES—A Real–Time Intrusion–Detection Expert System," SRI Project 6169, SRI International, Menlo Park, CA, Aug. 1985.

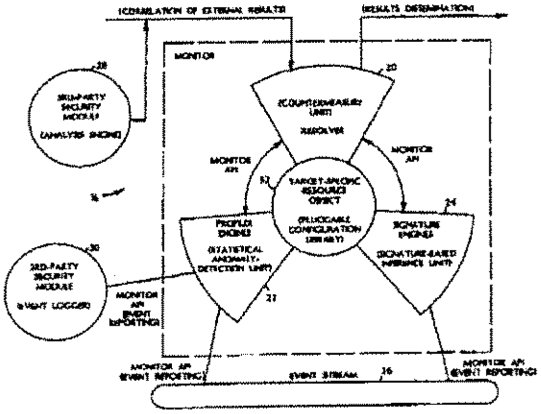
(List continued on next page.)

*Primary Examiner*—Thomas M. Heckler
(74) *Attorney, Agent, or Firm*—Moser, Patterson & Sheridan, LLP.; Kin-Wah Tong, Esq.

(57) **ABSTRACT**

A method of network surveillance includes receiving network packets handled by a network entity and building at least one long-term and a least one short-term statistical profile from a measure of the network packets that monitors data transfers, errors, or network connections. A comparison of the statistical profiles is used to determine whether the difference between the statistical profiles indicates suspicious network activity.

**93 Claims, 5 Drawing Sheets**


(CORRELATION OF EXTERNAL RESULTS)
(RESULTS DISSEMINATION)
MONITOR
28
3RD-PARTY SECURITY MODULE
(ANALYSIS ENGINE)
20
(COUNTERMEASURE) UNIT
RESOLVER
MONITOR API
MONITOR API
32
TARGET-SPECIFIC RESOURCE OBJECT
(PLUGGABLE CONFIGURATION LIBRARY)
16
24
PROFILER ENGINE
(STATISTICAL ANOMALY-DETECTION UNIT)
SIGNATURE ENGINE
(SIGNATURE-BASED INFERENCE UNIT)
30
3RD-PARTY SECURITY MODULE
(EVENT LOGGER)
MONITOR API (EVENT REPORTING)
22
MONITOR API (EVENT REPORTING)
EVENT STREAM
26
MONITOR API (EVENT REPORTING)

## U.S. PATENT DOCUMENTS

| | | | | |
|---|---|---|---|---|
| 5,790,799 | A | 8/1998 | Mogul | 709/224 |
| 5,878,420 | A | 3/1999 | de la Salle | 707/10 |
| 5,919,258 | A | 7/1999 | Kayashima et al. | 713/201 |
| 5,922,051 | A | 7/1999 | Sidey | 709/223 |
| 5,940,591 | A | 8/1999 | Boyle et al. | 395/187.01 |
| 5,974,237 | A | 10/1999 | Shurmer et al. | 709/224 |
| 5,974,457 | A | 10/1999 | Waclawshy et al. | 709/224 |
| 5,991,881 | A | 11/1999 | Conklin et al. | 713/201 |
| 6,009,467 | A | 12/1999 | Ratcliff et al. | 709/224 |
| 6,052,709 | A | 4/2000 | Paul | 709/202 |
| 6,070,244 | A | 5/2000 | Orchier et al. | 713/201 |
| 6,144,961 | A | 11/2000 | de la Salle | 707/10 |
| 6,396,845 | B1 | 5/2002 | Sugita | 709/224 X |
| 6,453,346 | B1 | 9/2002 | Garg et al. | 709/224 |
| 6,460,141 | B1 | 10/2002 | Olden | 712/201 |
| 6,519,703 | B1 | 2/2003 | Joyce | 713/201 |
| 2002/0032717 | A1 | 3/2002 | Malan et al. | 709/105 |
| 2002/0032793 | A1 | 3/2002 | Malan et al. | 709/232 |
| 2002/0032880 | A1 | 3/2002 | Poletto et al. | 714/4 |
| 2002/0035698 | A1 | 3/2002 | Malan et al. | 713/201 |
| 2002/0138753 | A1 | 9/2002 | Munson | 713/200 |
| 2002/0144156 | A1 | 10/2002 | Copeland, III | 713/201 |
| 2003/0037136 | A1 | 2/2003 | Labovitz et al. | 709/224 |

## OTHER PUBLICATIONS

Denning, "An Intrusion–Detection Model," SRI International, Menlo Park, CA Technical Report CSL–149, Nov. 1985.

Dowell, "The Computerwatch Data Reduction Tool," AT&T Bell Laboratories, Whippany, New Jersey.

Fox, et al., "A Neural Network Approach Towards Intrusion Detection," Harris Corporation, Government Information Systems Division, Melbourne, FL, Jul. 2, 1990.

Garvey, et al., "Model–Based Intrusion Detection," Proceedings of the 14th national Computer Security Conference, Washington, DC, Oct. 1991.

Garvey, et al., "An Inference Technique for Integrating Knowledge from Disparate Sources," Proc. IJCAI, Vancouver, BC, Aug. 1981, 319–325.

Ilgun et al., State Transition Analysis: A Rule–Based Intrusion Detection Approach, IEEE Transactions on Software Engineering, vol., 21, No. 3, Mar. 1995.

Javitz et al., "The SRI IDES Statistical Anomaly Detector," Proceedings, 1991 IEEE Symposium on Security and Privacy, Oakland, California, May 1991.

Jarvis et al., The NIDES Statistical Component Description and Justification, SRI International Annual Report A010, Mar. 7, 1994.

Kaven, "The Digital Dorman," PC Magazine, Nov. 16, 1999.

Liepins, et al., "Anomaly Detection; Purpose and Framework," US DOE Office of Safeguards and Security.

Lindquist, et al., "Detecting Computer and Network Misuse Through the Production–Based Expert System Toolset (P–BEST)," Oct. 25, 1998.

Lunt et al., "An Expert System to Classify and Sanitize Text," SRI International, Computer Science Laboratory, Menlo Park, CA.

Lunt, "A Survey of Intrusion Detection Techniques," Computers & Security, 12 (1993) 405–418

Lunt, "Automated Audit Trail Analysis and Intrusion Detection: A Survey," Proceedings of the 11th National Computer Security Conference, Baltimore, MD, Oct. 1988.

Lunt et al., Knowledge–Based Intrusion Detection Expert System, Proceedings of the 1988 IEEE Symposium on Security and Privacy, Apr. 1988.

Porras et al, "EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances," 20th NISSC—Oct. 9, 1997.

Porras et al., Penetration State Transition Analysis A Rule–Based Intrusion Detection Approach,© 1992 IEEE.

Sebring et al., Expert Systems in Intrusion Detection: A Case Study.

Shieh et al., A Pattern–Oriented Intrusion–Detection Model and Its Application © 1991 IEEE.

Smaha, Haystack: An Intrusion Detection System: © 1988 IEEE Computer Society Press: Proceedings of the Fourth Aerospace Computer Security Application Conference, 1988, pp. 37–44.

Snapp, Signature Analysis and Communication Issues in a Distributed Intrusion Detection System,: Thesis 1991.

Snapp et al., "DIDS (Distributed Intrusion Detection System)—Motivation, Architecture and An Early Prototype," Computer Security Laboratory, Division of Computer Science, Unic. Of California, Davis, Davis, CA.

Tener, "AI & 4GL: Automated Detection and Investigation Tools," Computer Security in the Age of Information, Proceedings of the Fifth IFIP International Conference on Computer Security, W.J. Caelli (ed.).

Teng et al., "Adaptive Real–Time Anomaly Detection Using Inductively Generated Sequential Patterns," © 1990.

Vaccaro et al., "Detection of Anomalous Computer Session Activity," © 1989 IEEE.

Weiss, "Analysis of Audit and Protocol Data using Methods from Artificial Intelligence," Siemens AG, Munich, West Germany.

Winkler, "A UNIX Prototype for Intrusion and Anomaly Detection in Secure Networks," Planning Research Corp. 1990.

Hartley, B., "Intrusion Detection Systems: What You Need to Know," Business Security Advisor Magazine, Doc # 05257, allegedly dated Sep. 1998, advisor.com/doc/05257, 7 pages, printed Jun. 10, 2003.

Hurwicz, M., "Cracker Tracking: Tighter Security with Intrusion Detection," BYTE.com, allegedly dated May 1998, www.byte.com/art/9805/sec20/art1.htm, 8 pages, printed Jun. 10, 2003.

"Networkers, Intrusion Detection and Scanning with Active Audit," Session 1305, © 1998 Cisco Systems, www.cisco.com/networkers/nw99 pres/1305.pdf, 0893–04F9_c3.scr, printed Jun. 10, 2003.

Paller, A., "About the SHADOW Intrusion Detection System" Linux Weekly News, allegedly dated Sep. 1998, lwn.net/1998/0910/shadow.html, 38 pages, printed Jun. 10, 2003.

Cisco Secure Intrusion Detection System, Release 2.1.1, NetRanger User's Guide, Version 2.1.1, © 1998, Cisco Systems, Inc., allegedly released on Apr. 1998, www.cisco.com/univercd/cc/td/doc/product/iaabu/csids/csids3/index.htm, printed Jun. 10, 2003, 334 pages, (See CSI document listed at C7 below).

Cisco Secure Intrusion Detection System 2.1.1 Release Notes, Table of Contents, Release Notes for NetRanger 2.1.1, © 1992–2002, Cisco Systems, Inc., , allegedly posted Sep. 28, 2002, 29 pages, www.cisco.com/univercd/cc/td/doc/product/iaabu/csids/csids3/nr11new.htm, printed Jun. 10, 2003.

R. Power, et al., "CSI Intrusion Detection System Resource", allegedly dated Jul. 1998, 216.239.57.100/search?q=cache:gvTCojxD6nMJ:www.gocsi.com/ques.htm+site:www.gocsi.com+ques&hl=en&ie=UTF-8, printed Jun. 16, 2003.

Lunt et al., "A Prototype Real-Time Intrusion-Detection Expert System," Proceedings of the 1988 IEEE Symposium on Security and Privacy, Apr. 1988.

Boyen, et al., "Tractable Inference for Complex Stochastic Processes," Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence (UAI-98), p. 33-42, Madison, WI, Jul. 24-26, 1998.

Copeland, J., "Observing Network Traffic—Techniques to Sort Out the Good, the Bad, and the Ugly," www.csc.gatech.edu/~copeland/8843/slides/Analyst-011027.ppt, allegedly 2001.

Farshchi, J., "Intrusion Detection FAQ, Statistical based approach to Intrusion Detection," www.sans.org/resources/idfaq/statistic ids.php, date unknown, printed 7/10/2003.

Goan, T., "A Cop on The Beat, Collecting and Appraising Intrusion Evidence," Communication of the ACM, 42(7), Jul. 1999, 46-52.

Heberlein, et al., "A Network Security Monitor," Proceedings of the IEEE Symposium on Security and Privacy, May 07-09 1990, Oakland, CA, pp 296-304, IEEE Press.

Internet Security Systems, "Intrusion Detection for the Millennium," ISS Technology Brief, Date Unknown, p. 1-6.

Jackson, et al., "An Expert System Application For Network Intrusion Detection," Proceedings of the 14th National Computer Security Conference, Washington, DC, 1-4 Oct. 1991.

Lankewicz, et al., "Real-time Anomaly Detection Using a Nonparametric Pattern Recognition Approach", Proceedings of the 7th Annual Computer Security Applications Conference, San Antonio, Texas, 1991, IEEE Press.

Lippmann, et al., "Evaluating Intrusion Detection Systems: The 1998 DARPA Off-line Intrusion Detection Evaluation," Proceedings of the 2000 DARPA, Information Survivability Conference and Exposition, Jan. 25-27 2000, Hilton Head, SC, vol. 2, pp 1012-1035, IEEE Press.

Miller, L., "A Network Under Attack, Leverage Your Existing Instrumentation to Recognize and Respond to Hacker Attacks," www.netscout.com/files/Intrusion 020118.pdf, Date Unknown, p. 1-8.

Munson, et al., "Watcher: The Missing Piece of the Security Puzzle," Proceedings of the 17th Annual Computer Security Applications Conference (ACSAC'01), Dec. 10-14 2001, New Orleans, LA, pp 230-239, IEEE Press.

NetScreen, Products FAQ, www.netscreen.com/products/faq.html, Date Unknown.

Pearl, J., "Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference," Morgan Kaufmann Publishers, Sep. 1988.

Porras, et al., "Live Traffic Analysis of TCP/IP Gateways," Proc. 1998 ISOC Symp. On Network and Distributed Systems Security, Dec. 12, 1997, 1-13.

Skinner, "EMERALD TCP Statistical Analyzer 1998 Evaluation Results," www.sdl.sri.com/emerald/98-eval-estat/index.html, Allegedly dated Jul. 9, 1999.

SRI/Stanford, "Adaptive Model-Based Monitoring and Threat Detection," Information Assurance BAA 98-34.

Staniford-Chen, et al., "GrIDS—A Graph Based Intrusion Detection System for Large Networks," Proceedings of the 19th National Information Systems Security Conference, vol. 1, pp 361-370, Oct. 1996.

Tener, "Discovery: An Expert System in the Commercial Data Security Environment", Fourth IFIP Symposium on Information Systems Security, Monte Carlo, Dec. 1986.

Valdes, et al., "Adaptive, Model-based Monitoring for Cyber Attack Detection," Proceedings of Recent Advances in Intrusion Detection 2000 (RAID 2000), H. Debar, L. Me, F. Wu (Eds), Toulouse, France, Springer-Verlag LNCS vol. 1907, pp 80-92, Oct. 2000.

Valdes, A., Blue Sensors, Sensor Correlation, and Alert Fusion, www.raid-symposium.org/raid2000/Materials/Abstracts/41/avaldes raidB.pdf, Oct. 4, 2000.

Valdes, et al., "Statistical Methods for Computer Usage Anomaly Detection Using NIDES (Next-Generation Intrusion Detection Expert System)," 3rd International Workshop on Rough Sets and Soft Computing, San Jose CA 1995, 306-311.

Wimer, S., "The Core of CylantSecure," White Papers, www.cylant.com/products/core.html, Date Unknown, Alleged © 1999-2003 Cylant Inc., pp. 1-4.

Zhang, et al., "A Hierarchical Anomaly Network Intrusion Detection System using Neural Network Classification," Proceedings of the 2001 WSES International Conference on Neural Networks and Applications (NNA'01), Puerto de la Cruz, Canary Islands, Spain, Feb. 11-15 2001.

(12) **United States Patent**
**Porras et al.**

(10) **Patent No.: US 6,711,615 B2**
(45) **Date of Patent: *Mar. 23, 2004**

(54) **NETWORK SURVEILLANCE**

(75) Inventors: **Phillip Andrew Porras**, Moutain View, CA (US); **Alfonso Valdes**, San Carlos, CA (US)

(73) Assignee: **SRI International**, Menlo Park, CA (US)

( * ) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **10/254,457**

(22) Filed: **Sep. 25, 2002**

(65) **Prior Publication Data**

US 2003/0088791 A1 May 8, 2003

**Related U.S. Application Data**

(63) Continuation of application No. 09/658,137, filed on Sep. 8, 2000, now Pat. No. 6,484,203, which is a continuation of application No. 09/188,739, filed on Nov. 9, 1998, now Pat. No. 6,321,338.

(51) **Int. Cl.**$^7$ .......................... **G06F 11/30**; G06F 12/14
(52) **U.S. Cl.** .......................... **709/224**; 713/201
(58) **Field of Search** .......................... 713/200, 201; 709/223–225

(56) **References Cited**

U.S. PATENT DOCUMENTS

| | | | | |
|---|---|---|---|---|
| 4,672,609 A | 6/1987 | Humphrey et al. | .......... | 371/21 |
| 4,773,028 A | 9/1988 | Tallman | .......... | 364/550 |
| 5,210,704 A | 5/1993 | Husseiny | .......... | 364/551.01 |
| 5,440,723 A | 8/1995 | Arnold et al. | .......... | 395/181 |
| 5,539,659 A | 7/1996 | McKee et al. | .......... | 709/224 |
| 5,557,742 A | 9/1996 | Smaha et al. | .......... | 395/186 |
| 5,706,210 A | 1/1998 | Kumano et al. | .......... | 709/224 |
| 5,748,098 A | 5/1998 | Grace | .......... | 340/825.16 |

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

| | | | |
|---|---|---|---|
| WO | 99/13427 | 3/1999 | .......... G06K/7/00 |
| WO | 99/57626 | 11/1999 | .......... G06F/1/16 |
| WO | 00/10278 | 2/2000 | |
| WO | 00/25214 | 5/2000 | .......... G06F/12/14 |
| WO | 00/25527 | 5/2000 | .......... H04Q/3/00 |
| WO | 00/34867 | 6/2000 | .......... G06F/11/30 |
| WO | 02/101516 | 12/2002 | |

OTHER PUBLICATIONS

Debar, et al., "Towards a Taxonomy of Intrusion–Detection Systems," Computer Networks 31 (1999), 805–822.

Debar et al., "A Neural Network Component for an Intrusion Detection System," © 1992 IEEE.

Denning et al, "Prototype IDES: A Real–Time Intrusion–Detection Expert System," SRI Project ECU 7508, SRI International, Menlo Park, California, Aug. 1987.

Denning et al., "Requirements and Model for IDES—A Real–Time Intrusion–Detection Expert System," SRI Project 6169, SRI International, Menlo Park, CA, Aug. 1985.

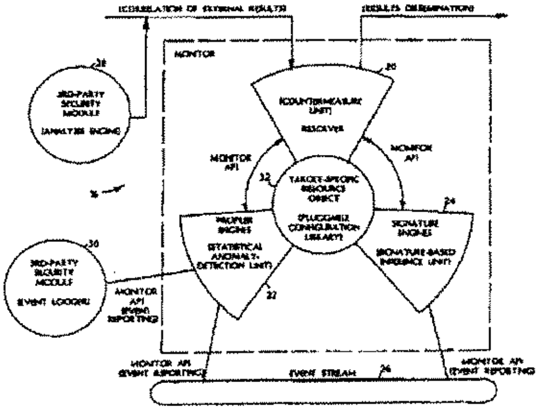
(List continued on next page.)

*Primary Examiner*—Thomas M. Heckler
(74) *Attorney, Agent, or Firm*—Moser, Patterson & Sheridan, LLP.; Kin-Wah Tong, Esq.

(57) **ABSTRACT**

A method of network surveillance includes receiving network packets handled by a network entity and building at least one long-term and a least one short-term statistical profile from a measure of the network packets that monitors data transfers, errors, or network connections. A comparison of the statistical profiles is used to determine whether the difference between the statistical profiles indicates suspicious network activity.

**93 Claims, 5 Drawing Sheets**


(CORRELATION OF EXTERNAL RESULTS)
(RESULTS DISSEMINATION)
MONITOR
28
3RD-PARTY SECURITY MODULE
(ANALYSIS ENGINE)
20
(COUNTERMEASURE UNIT)
RESOLVER
MONITOR API
MONITOR API
32
TARGET-SPECIFIC RESOURCE OBJECT
(PLUGGABLE CONFIGURATION LIBRARY)
16
24
PROFILER ENGINES
(STATISTICAL ANOMALY-DETECTION UNIT)
22
SIGNATURE ENGINES
(SIGNATURE-BASED INFERENCE UNIT)
30
3RD-PARTY SECURITY MODULE
(EVENT LOGGER)
MONITOR API (EVENT REPORTING)
MONITOR API (EVENT REPORTING)
EVENT STREAM
26
MONITOR API (EVENT REPORTING)

## U.S. PATENT DOCUMENTS

| | | | | |
|---|---|---|---|---|
| 5,790,799 | A | 8/1998 | Mogul | 709/224 |
| 5,878,420 | A | 3/1999 | de la Salle | 707/10 |
| 5,919,258 | A | 7/1999 | Kayashima et al. | 713/201 |
| 5,922,051 | A | 7/1999 | Sidey | 709/223 |
| 5,940,591 | A | 8/1999 | Boyle et al. | 395/187.01 |
| 5,974,237 | A | 10/1999 | Shurmer et al. | 709/224 |
| 5,974,457 | A | 10/1999 | Waclawshy et al. | 709/224 |
| 5,991,881 | A | 11/1999 | Conklin et al. | 713/201 |
| 6,009,467 | A | 12/1999 | Ratcliff et al. | 709/224 |
| 6,052,709 | A | 4/2000 | Paul | 709/202 |
| 6,070,244 | A | 5/2000 | Orchier et al. | 713/201 |
| 6,144,961 | A | 11/2000 | de la Salle | 707/10 |
| 6,396,845 | B1 | 5/2002 | Sugita | 709/224 X |
| 6,453,346 | B1 | 9/2002 | Garg et al. | 709/224 |
| 6,460,141 | B1 | 10/2002 | Olden | 712/201 |
| 6,519,703 | B1 | 2/2003 | Joyce | 713/201 |
| 2002/0032717 | A1 | 3/2002 | Malan et al. | 709/105 |
| 2002/0032793 | A1 | 3/2002 | Malan et al. | 709/232 |
| 2002/0032880 | A1 | 3/2002 | Poletto et al. | 714/4 |
| 2002/0035698 | A1 | 3/2002 | Malan et al. | 713/201 |
| 2002/0138753 | A1 | 9/2002 | Munson | 713/200 |
| 2002/0144156 | A1 | 10/2002 | Copeland, III | 713/201 |
| 2003/0037136 | A1 | 2/2003 | Labovitz et al. | 709/224 |

## OTHER PUBLICATIONS

Denning, "An Intrusion-Detection Model," SRI International, Menlo Park, CA Technical Report CSL-149, Nov. 1985.

Dowell, "The Computerwatch Data Reduction Tool," AT&T Bell Laboratories, Whippany, New Jersey.

Fox, et al., "A Neural Network Approach Towards Intrusion Detection," Harris Corporation, Government Information Systems Division, Melbourne, FL, Jul. 2, 1990.

Garvey, et al., "Model-Based Intrusion Detection," Proceedings of the 14th national Computer Security Conference, Washington, DC, Oct. 1991.

Garvey, et al., "An Inference Technique for Integrating Knowledge from Disparate Sources," Proc. IJCAI, Vancouver, BC, Aug. 1981, 319-325.

Ilgun et al., State Transition Analysis: A Rule-Based Intrusion Detection Approach, IEEE Transactions on Software Engineering, vol., 21, No. 3, Mar. 1995.

Javitz et al., "The SRI IDES Statistical Anomaly Detector," Proceedings, 1991 IEEE Symposium on Security and Privacy, Oakland, California, May 1991.

Jarvis et al., The NIDES Statistical Component Description and Justification, SRI International Annual Report A010, Mar. 7, 1994.

Kaven, "The Digital Dorman," PC Magazine, Nov. 16, 1999.

Liepins, et al., "Anomaly Detection; Purpose and Framework," US DOE Office of Safeguards and Security.

Lindquist, et al., "Detecting Computer and Network Misuse Through the Production-Based Expert System Toolset (P-BEST)," Oct. 25, 1998.

Lunt et al., "An Expert System to Classify and Sanitize Text," SRI International, Computer Science Laboratory, Menlo Park, CA.

Lunt, "A Survey of Intrusion Detection Techniques," Computers & Security, 12 (1993) 405-418.

Lunt, "Automated Audit Trail Analysis and Intrusion Detection: A Survey," Proceedings of the 11th National Computer Security Conference, Baltimore, MD, Oct. 1988.

Lunt et al., Knowledge-Based Intrusion Detection Expert System, Proceedings of the 1988 IEEE Symposium on Security and Privacy, Apr. 1988.

Porras et al, "EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances," 20th NISSC—Oct. 9, 1997.

Porras et al., Penetration State Transition Analysis A Rule-Based Intrusion Detection Approach,© 1992 IEEE.

Sebring et al., Expert Systems in Intrusion Detection: A Case Study.

Shieh et al., A Pattern-Oriented Intrusion-Detection Model and Its Application © 1991 IEEE.

Smaha, Haystack: An Intrusion Detection System: © 1988 IEEE Computer Society Press: Proceedings of the Fourth Aerospace Computer Security Application Conference, 1988, pp. 37-44.

Snapp, Signature Analysis and Communication Issues in a Distributed Intrusion Detection System,: Thesis 1991.

Snapp et al., "DIDS (Distributed Intrusion Detection System)—Motivation, Architecture and An Early Prototype," Computer Security Laboratory, Division of Computer Science, Unic. Of California, Davis, Davis, CA.

Tener, "AI & 4GL: Automated Detection and Investigation Tools," Computer Security in the Age of Information, Proceedings of the Fifth IFIP International Conference on Computer Security, W.J. Caelli (ad.).

Teng et al., "Adaptive Real-Time Anomaly Detection Using Inductively Generated Sequential Patterns," © 1990.

Vaccaro et al., "Detection of Anomalous Computer Session Activity," © 1989 IEEE.

Weiss, "Analysis of Audit and Protocol Data using Methods from Artificial Intelligence," Siemens AG, Munich, West Germany.

Winkler, "A UNIX Prototype for Intrusion and Anomaly Detection in Secure Networks," Planning Research Corp. 1990.

Hartley, B., "Intrusion Detection Systems: What You Need to Know," Business Security Advisor Magazine, Doc # 05257, allegedly dated Sep. 1998, advisor.com/doc/05257, 7 pages, printed Jun. 10, 2003.

Hurwicz, M., "Cracker Tracking: Tighter Security with Intrusion Detection," BYTE.com, allegedly dated May 1998, www.byte.com/art/9805/sec20/art1.htm, 8 pages, printed Jun. 10, 2003.

"Networkers. Intrusion Detection and Scanning with Active Audit," Session 1305, © 1998 Cisco Systems, www.cisco-.com/networkers/nw99 pres/1305.pdf, 0893-04F9_c3.scr, printed Jun. 10, 2003.

Paller, A., "About the SHADOW Intrusion Detection System" Linux Weekly News, allegedly dated Sep. 1998, lwn-.net/1998/0910/shadow.html, 38 pages, printed Jun. 10, 2003.

Cisco Secure Intrusion Detection System, Release 2.1.1, NetRanger User's Guide, Version 2.1.1, © 1998, Cisco Systems, Inc., allegedly released on Apr. 1998, www.cisco-.com/univercd/cc/td/doc/product/iaabu/csids/csids3/index.htm, printed Jun. 10, 2003, 334 pages, (See CSI document listed at C7 below).

Cisco Secure Intrusion Detection System 2.1.1 Release Notes, Table of Contents, Release Notes for NetRanger 2.1.1, © 1992-2002, Cisco Systems, Inc., , allegedly posted Sep. 28, 2002, 29 pages, www.cisco.com/univercd/cc/td/doc/product/iaabu/csids/csids3/nr11new.htm, printed Jun. 10, 2003.

R. Power, et al., "CSI Intrusion Detection System Resource", allegedly dated Jul. 1998, 216.239.57.100/search?q=cache:gvTCojxD6nMJ:www.gocsi.com/ques.htm+site:www.gocsi.com+ques&hl=en&ie=UTF–8, printed Jun. 16, 2003.

Lunt et al., "A Prototype Real–Time Intrusion–Detection Expert System," Proceedings of the 1988 IEEE Symposium on Security and Privacy, Apr. 1988.

Boyen, et al., "Tractable Inference for Complex Stochastic Processes," Proceedings of the $14^{th}$ Annual Conference on Uncertainty in Artificial Intelligence (UAI–98), p. 33–42, Madison, WI, Jul. 24–26, 1998.

Copeland, J., "Observing Network Traffic—Techniques to Sort Out the Good, the Bad, and the Ugly," www.csc.gatech.edu/~copeland/8843/slides/Analyst–011027.ppt, allegedly 2001.

Farshchi, J., "Intrusion Detection FAQ, Statistical based approach to Intrusion Detection," www.sans.org/resources/idfaq/statistic ids.php, date unknown, printed 7/10/2003.

Goan, T., "A Cop on The Beat, Collecting and Appraising Intrusion Evidence," Communication of the ACM, 42(7), Jul. 1999, 46–52.

Heberlein, et al., "A Network Security Monitor," Proceedings of the IEEE Symposium on Security and Privacy, May 07–09 1990, Oakland, CA, pp 296–304, IEEE Press.

Internet Security Systems, "Intrusion Detection for the Millennium," ISS Technology Brief, Date Unknown, p. 1–6.

Jackson, et al., "An Expert System Application For Network Intrusion Detection," Proceedings of the 14th National Computer Security Conference, Washington, DC, 1–4 Oct. 1991.

Lankewicz, et al., "Real–time Anomaly Detection Using a Nonparametric Pattern Recognition Approach", Proceedings of the 7th Annual Computer Security Applications Conference, San Antonio, Texas, 1991, IEEE Press.

Lippmann, et al., "Evaluating Intrusion Detection Systems: The 1998 DARPA Off–line Intrusion Detection Evaluation," Proceedings of the 2000 DARPA, Information Survivability Conference and Exposition, Jan. 25–27 2000, Hilton Head, SC, vol. 2, pp 1012–1035, IEEE Press.

Miller, L., "A Network Under Attack, Leverage Your Existing Instrumentation to Recognize and Respond to Hacker Attacks," www.netscout.com/files/Intrusion 020118.pdf, Date Unknown, p. 1–8.

Munson, et al., "Watcher: The Missing Piece of the Security Puzzle," Proceedings of the 17th Annual Computer Security Applications Conference (ACSAC'01), Dec. 10–14 2001, New Orleans, LA, pp 230–239, IEEE Press.

NetScreen, Products FAQ, www.netscreen.com/products/faq.html, Date Unknown.

Pearl, J., "Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference," Morgan Kaufmann Publishers, Sep. 1988.

Porras, et al., "Live Traffic Analysis of TCP/IP Gateways," Proc. 1998 ISOC Symp. On Network and Distributed Systems Security, Dec. 12, 1997, 1–13.

Skinner, "EMERALD TCP Statistical Analyzer 1998 Evaluation Results," www.sdl.sri.com/emerald/98–eval–estat/index.html, Allegedly dated Jul. 9, 1999.

SRI/Stanford, "Adaptive Model–Based Monitoring and Threat Detection," Information Assurance BAA 98–34.

Staniford–Chen, et al., "GrIDS—A Graph Based Intrusion Detection System for Large Networks," Proceedings of the 19th National Information Systems Security Conference, vol. 1, pp 361–370, Oct. 1996.

Tener, "Discovery: An Expert System in the Commercial Data Security Environment", Fourth IFIP Symposium on Information Systems Security, Monte Carlo, Dec. 1986.

Valdes, et al., "Adaptive, Model–based Monitoring for Cyber Attack Detection," Proceedings of Recent Advances in Intrusion Detection 2000 (RAID 2000), H. Debar, L. Me, F. Wu (Eds), Toulouse, France, Springer–Verlag LNCS vol. 1907, pp 80–92. Oct. 2000.

Valdes, A., Blue Sensors, Sensor Correlation, and Alert Fusion, www.raid–symposium.org/raid2000/Materials/Abstracts/41/avaldes raidB.pdf, Oct. 4, 2000.

Valdes, et al., "Statistical Methods for Computer Usage Anomaly Detection Using NIDES (Next–Generation Intrusion Detection Expert System)," 3rd International Workshop on Rough Sets and Soft Computing, San Jose CA 1995, 306–311.

Wimer, S., "The Core of CylantSecure," White Papers, www.cylant.com/products/core.html, Date Unknown, Alleged © 1999–2003 Cylant Inc., pp. 1–4.

Zhang, et al., "A Hierarchical Anomaly Network Intrusion Detection System using Neural Network Classification," Proceedings of the 2001 WSES International Conference on Neural Networks and Applications (NNA'01), Puerto de la Cruz, Canary Islands, Spain, Feb. 11–15 2001.

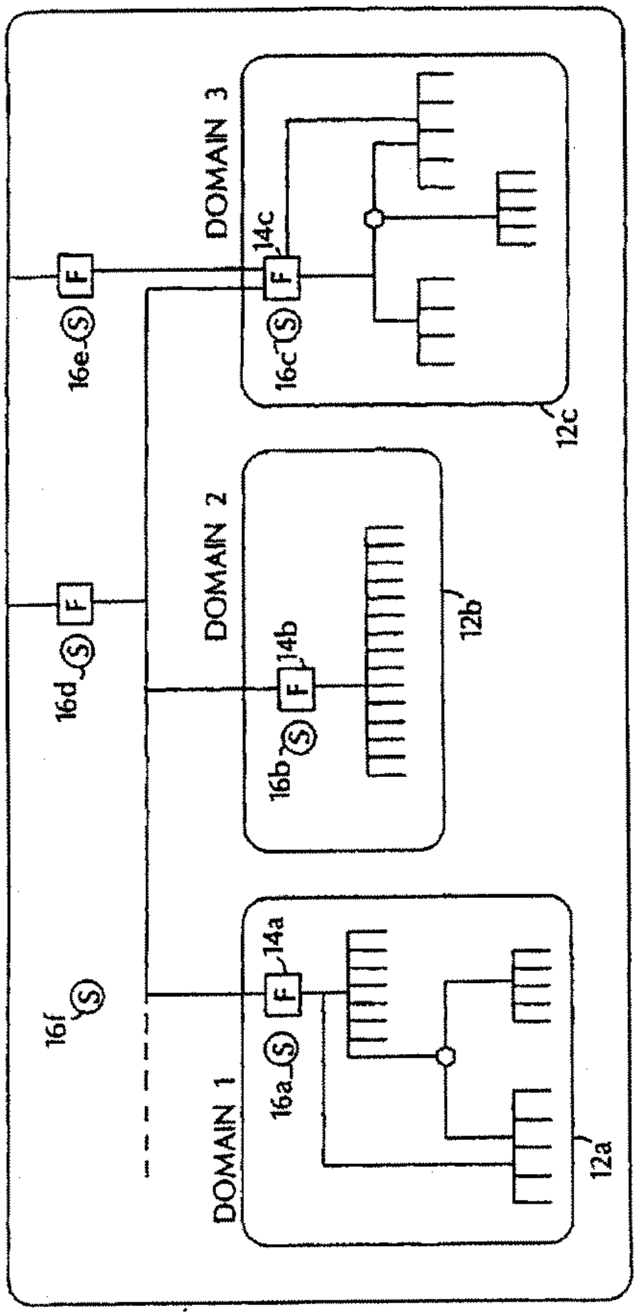

DOMAIN 3
14c
F
S
16c
16e
F
S
12c
DOMAIN 2
14b
F
S
16b
16d
F
S
12b
DOMAIN 1
14a
F
S
16a
16f
S
12a

FIG. 1

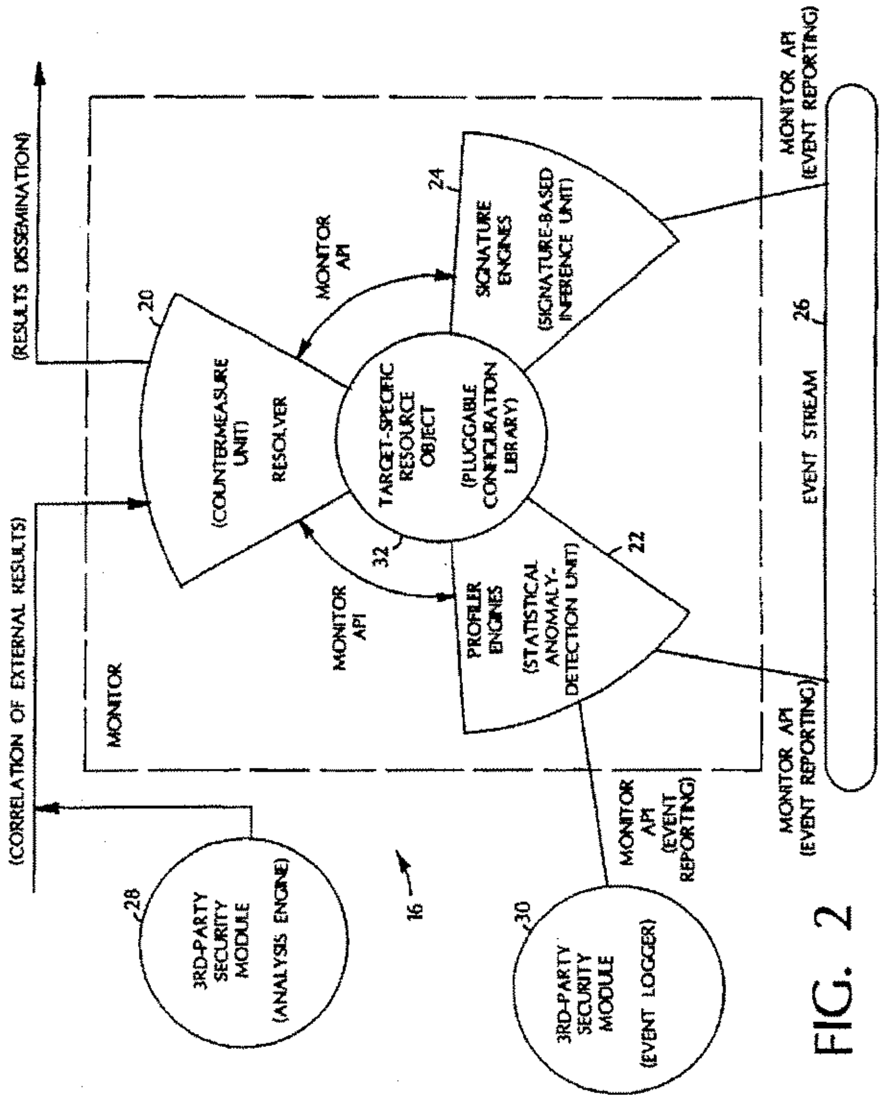

(RESULTS DISSEMINATION)
(CORRELATION OF EXTERNAL RESULTS)
MONITOR
20
(COUNTERMEASURE UNIT)
RESOLVER
MONITOR API
MONITOR API
TARGET-SPECIFIC RESOURCE OBJECT
(PLUGGABLE CONFIGURATION LIBRARY)
32
24
SIGNATURE ENGINES
(SIGNATURE-BASED INFERENCE UNIT)
22
PROFILER ENGINES
(STATISTICAL ANOMALY-DETECTION UNIT)
26
EVENT STREAM
MONITOR API (EVENT REPORTING)
MONITOR API (EVENT REPORTING)
MONITOR API (EVENT REPORTING)
28
3RD-PARTY SECURITY MODULE
(ANALYSIS ENGINE)
16
30
3RD-PARTY SECURITY MODULE
(EVENT LOGGER)

FIG. 2

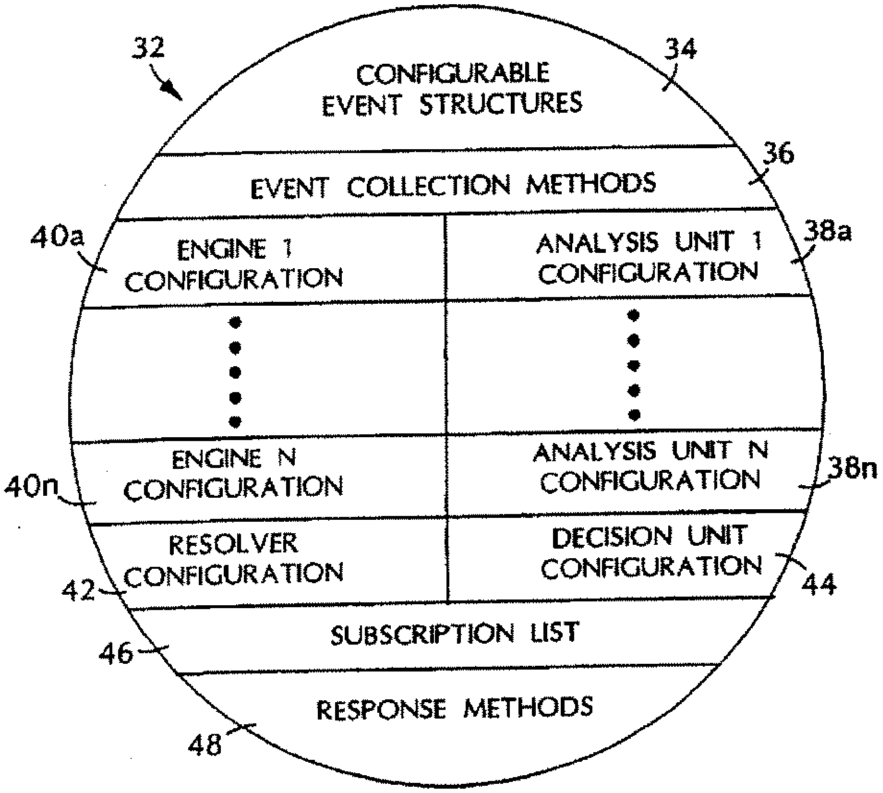

32
34
CONFIGURABLE EVENT STRUCTURES
36
EVENT COLLECTION METHODS
40a
ENGINE 1 CONFIGURATION
ANALYSIS UNIT 1 CONFIGURATION
38a
40n
ENGINE N CONFIGURATION
ANALYSIS UNIT N CONFIGURATION
38n
RESOLVER CONFIGURATION
DECISION UNIT CONFIGURATION
42
44
46
SUBSCRIPTION LIST
48
RESPONSE METHODS

FIG. 3

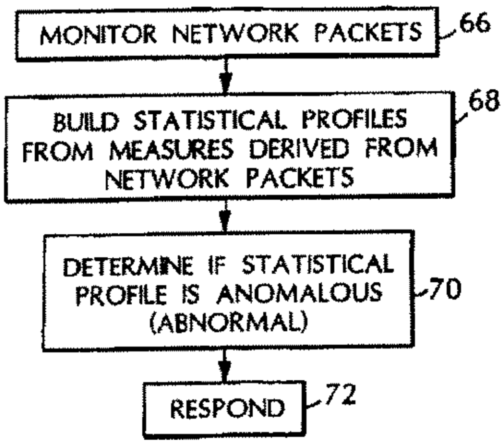

MONITOR NETWORK PACKETS
66
BUILD STATISTICAL PROFILES FROM MEASURES DERIVED FROM NETWORK PACKETS
68
DETERMINE IF STATISTICAL PROFILE IS ANOMALOUS (ABNORMAL)
70
RESPOND
72

FIG. 4

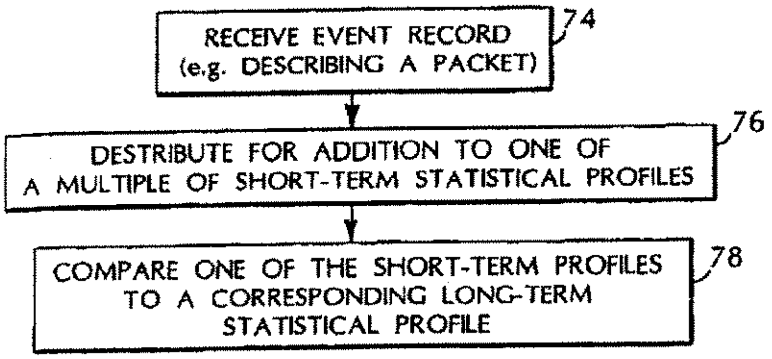

RECEIVE EVENT RECORD (e.g. DESCRIBING A PACKET)
74
DESTRIBUTE FOR ADDITION TO ONE OF A MULTIPLE OF SHORT-TERM STATISTICAL PROFILES
76
COMPARE ONE OF THE SHORT-TERM PROFILES TO A CORRESPONDING LONG-TERM STATISTICAL PROFILE
78

FIG. 5

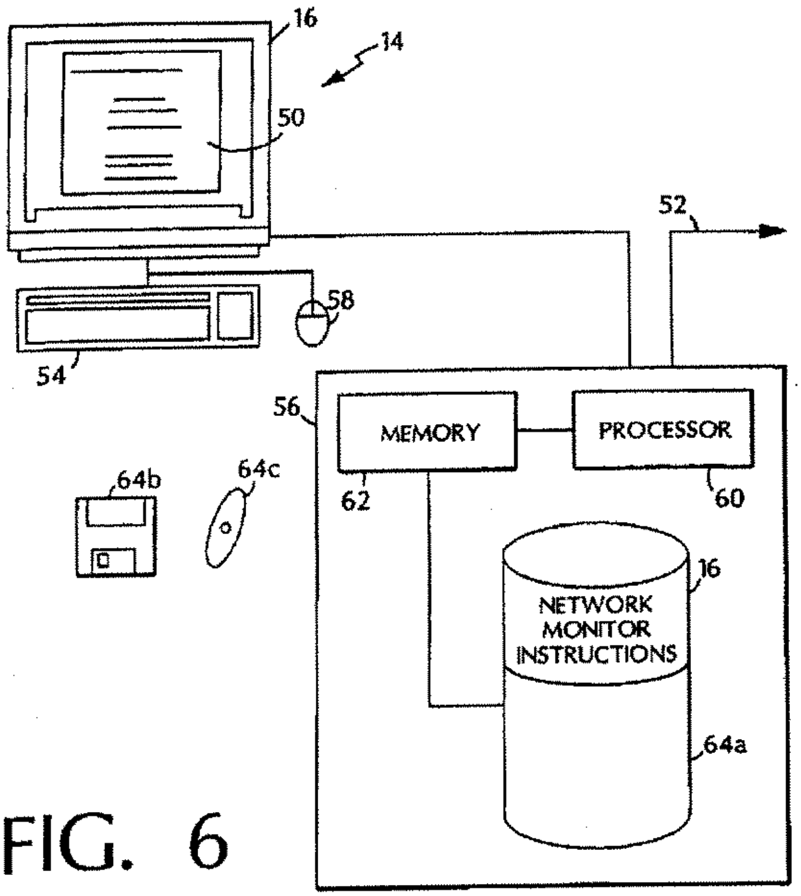

16
14
50
52
58
54
56
MEMORY
PROCESSOR
62
60
64b
64c
16
NETWORK MONITOR INSTRUCTIONS
64a

FIG. 6

## NETWORK SURVEILLANCE

This application is a continuation of U.S. application Ser. No. 09/658,137, filed on Sep. 8, 2000 (now U.S. Pat. No. 6,484,203), which is a continuation of U.S. application Ser. No. 09/188,739, filed Nov. 9, 1998 (now U.S. Pat. No. 6,321,338), where both applications are herein incorporated by reference.

### REFERENCE TO GOVERNMENT FUNDING

This invention was made with Government support under Contract Number F30602-96-C-0294 and F30602-96-C-0187 awarded by DARPA and the Air Force Research Laboratory. The Government has certain rights in this invention.

### REFERENCE TO APPENDIX

An appendix consisting of 935 pages is included as part of the specification. The appendix includes material subject to copyright protection. The copyright owner does not object to the facsimile reproduction of the appendix, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights.

### BACKGROUND

The invention relates to computer networks.

Computer networks offer users ease and efficiency in exchanging information. Networks tend to include conglomerates of integrated commercial and custom-made components, interoperating and sharing information at increasing levels of demand and capacity. Such varying networks manage a growing list of needs including transportation, commerce, energy management, communications, and defense.

Unfortunately, the very interoperability and sophisticated integration of technology that make networks such valuable assets also make them vulnerable to attack, and make dependence on networks a potential liability. Numerous examples of planned network attacks, such as the Internet worm, have shown how interconnectivity can be used to spread harmful program code. Accidental outages such as the 1980 ARPAnet collapse and the 1990 AT&T collapse illustrate how seemingly localized triggering events can have globally disastrous effects on widely distributed systems. In addition, organized groups have performed malicious and coordinated attacks against various online targets.

### SUMMARY

In general, in one aspect, a method of network surveillance includes receiving network packets (e.g., TCP/IP packets) handled by a network entity and building at least one long-term and at least one short-term statistical profile from at least one measure of the network packets that monitors data transfers, errors, or network connections. A comparison of at least one long-term and at least one short-term statistical profile is used to determine whether the difference between the short-term statistical profile and the long-term statistical profile indicates suspicious network activity.

Embodiments may include one or more of the following features. The measure may monitor data transfers by monitoring network packet data transfer commands, data transfer errors, and/or monitoring network packet data transfer volume. The measure may monitor network connections by monitoring network connection requests, network connection denials, and/or a correlation of network connections requests and network connection denials. The measure may monitor errors by monitoring error codes included in a network packet such as privilege error codes and/or error codes indicating a reason a packet was rejected.

The method may also include responding based on the determining whether the difference between a short-term statistical profile and a long-term statistical profile indicates suspicious network activity. A response may include altering analysis of network packets and/or severing a communication channel. A response may include transmitting an event record to a network monitor, such as hierarchically higher network monitor and/or a network monitor that receives event records from multiple network monitors.

The network entity may be a gateway, a router, or a proxy server. The network entity may instead be a virtual private network entity (e.g., node).

In general, in another aspect, a method of network surveillance includes monitoring network packets handled by a network entity and building a long-term and multiple short-term statistical profiles of the network packets. A comparison of one of the multiple short-term statistical profiles with the long-term statistical profile is used to determine whether the difference between the short-term statistical profiles and the long-term statistical profile indicates suspicious network activity.

Embodiments may include one or more of the following. The multiple short-term statistical profiles may monitor different anonymous FTP sessions. Building multiple short-term statistical profiles may include deinterleaving packets to identify a short-term statistical profile.

In general, in another aspect, a computer program product, disposed on a computer readable medium, includes instructions for causing a processor to receive network packets handled by a network entity and to build at least one long-term and at least one short-term statistical profile from at least one measure of the network packets that monitors data transfers, errors, or network connections. The instructions compare a short-term and a long-term statistical profile to determine whether the difference between the short-term statistical profile and the long-term statistical profile indicates suspicious network activity.

In general, in another aspect, a method of network surveillance includes receiving packets at a virtual private network entity and statistically analyzing the received packets to determine whether the packets indicate suspicious network activity. The packets may or may not be decrypted before statistical analysis

Advantages may include one or more of the following. Using long-term and a short-term statistical profiles from measures that monitor data transfers, errors, or network connections protects network components from intrusion. As long-term profiles represent "normal" activity, abnormal activity may be detected without requiring an administrator to catalog each possible attack upon a network. Additionally, the ability to deinterleave packets to create multiple short-term profiles for comparison against a long-term profile enables the system to detect abnormal behavior that may be statistically ameliorated if only a single short-term profile was created.

The scheme of communication network monitors also protects networks from more global attacks. For example, an attack made upon one network entity may cause other entities to be alerted. Further, a monitor that collects event reports from different monitors may correlate activity to identify attacks causing disturbances in more than one network entity.

Additionally, statistical analysis of packets handled by a virtual private network enable detection of suspicious network activity despite virtual private network security techniques such as encryption of the network packets.

Other features and advantages will become apparent from the following description, including the drawings, and from the claims.

## BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a diagram of network monitors deployed in an enterprise.

FIG. 2 is a diagram of a network monitor that monitors an event stream.

FIG. 3 is a diagram of a resource object that configures the network monitor of FIG. 2.

FIG. 4 is a flowchart illustrating network surveillance.

FIG. 5 is a flowchart illustrating multiple short-term statistical profiles for comparison against a single long-term statistical profile.

FIG. 6 is a diagram of a computer platform suitable for deployment of a network monitor.

## DETAILED DESCRIPTION

Referring to FIG. 1, an enterprise 10 includes different domains 12a–12c. Each domain 12a–12c includes one or more computers offering local and network services that provide an interface for requests internal and external to the domain 12a–12c. Network services include features common to many network operating systems such as mail, HTTP, FTP, remote login, network file systems, finger, Kerberos, and SNMP. Some domains 12a–12c may share trust relationships with other domains (either peer-to-peer or hierarchical). Alternatively, domains 12a–12c may operate in complete mistrust of all others, providing outgoing connections only or severely restricting incoming connections. Users may be local to a single domain or may possess accounts on multiple domains that allow them to freely establish connections throughout the enterprise 10.

As shown, the enterprise 10 includes dynamically deployed network monitors 16a–16f that analyze and respond to network activity and can interoperate to form an analysis hierarchy. The analysis hierarchy provides a framework for the recognition of more global threats to interdomain connectivity, including coordinated attempts to infiltrate or destroy connectivity across an entire network enterprise 10. The hierarchy includes service monitors 16a–16c, domain monitors 16d–16e, and enterprise monitors 16f.

Service monitors 16a–16c provide local real-time analysis of network packets (e.g., TCP/IP packets) handled by a network entity 14a–14c. Network entities include gateways, routers, firewalls, or proxy servers. A network entity may also be part of a virtual private network. A virtual private network (VPN) is constructed by using public wires to connect nodes. For example, a network could use the Internet as the medium for transporting data and use encryption and other security mechanisms to ensure that only authorized users access the network and that the data cannot be intercepted. A monitor 16a–16f can analyze packets both before and after decryption by a node of the virtual private network.

Information gathered by a service monitor 16a–16c can be disseminated to other monitors 16a–16f, for example, via a subscription-based communication scheme. In a subscription-based scheme client monitors subscribe to receive analysis reports produced by server monitors. As a monitor 16a–16f produces analysis reports, the monitor 16a–16f disseminates these reports asynchronously to subscribers. Through subscription, monitors 16a–16f distributed throughout a large network are able to efficiently disseminate reports of malicious activity without requiring the overhead of synchronous polling.

Domain monitors 16d–16e perform surveillance over all or part of a domain 12a–12c. Domain monitors 16d–16e correlate intrusion reports disseminated by individual service monitors 16a–16c, providing a domain-wide perspective of activity (or patterns of activity). In addition to domain surveillance, domain monitors 16a–16c can reconfigure system parameters, interface with other monitors beyond a domain, and report threats against a domain 12a–12c to administrators. Domain monitors 16d–16e can subscribe to service monitors 16a–16c. Where mutual trust among domains 12a–12c exists, domain monitors 16d–16e may establish peer relationships with one another. Peer-to-peer subscription allows domain monitors 16d–16e to share analysis reports produced in other domains 12a–12c. Domain monitors 16d–16e may use such reports to dynamically sensitize their local service monitors 16a–16c to malicious activity found to be occurring outside a domain 12a–12c. Domain monitors 16d–16e may also operate within an enterprise hierarchy where they disseminate analysis reports to enterprise monitors 16f for global correlation.

Enterprise monitors 16f correlate activity reports produced across the set of monitored domains 12a–12c. Enterprise 10 surveillance may be used where domains 12a–12c are interconnected under the control of a single organization, such as a large privately owned WAN (Wide Area Network). The enterprise 10, however, need not be stable in its configuration or centrally administered. For example, the enterprise 10 may exist as an emergent entity through new interconnections of domains 12a–12c. Enterprise 10 surveillance is very similar to domain 12a–12c surveillance: an enterprise monitor 16f subscribes to various domain monitors 16d–16e, just as the domain monitors 16d–16e subscribed to various service monitors 16a–16c. The enterprise monitor 16f (or monitors, as it would be important to avoid centralizing any analysis) focuses on network-wide threats such as Internet worm-like attacks, attacks repeated against common network services across domains, or coordinated attacks from multiple domains against a single domain. As an enterprise monitor 16f recognizes commonalities in intrusion reports across domains (e.g., the spreading of a worm or a mail system attack repeated throughout the enterprise 10), the monitor 16f can help domains 12a–12c counter the attack and can sensitize other domains 12a–12c to such attacks before they are affected. Through correlation and sharing of analysis reports, reports of problems found by one monitor 16a–16f may propagate to other monitors 16a–16f throughout the network. Interdomain event analysis is vital to addressing more global, information attacks against the entire enterprise 10.

Referring to FIG. 2, each monitor 16 includes one or more analysis engines 22, 24. These engines 22, 24 can be dynamically added, deleted, and modified as necessary. In the dual-analysis configuration shown, a monitor 16 instantiation includes a signature analysis engine 22 and a statistical profiling engine 24. In general, a monitor 16 may include additional analysis engines that may implement other forms of analysis. A monitor 16 also includes a resolver 20 that implements a response policy and a resource object 32 that configures the monitor 16. The monitors 16 incorporate an application programmers' interface (API) that enhances encapsulation of monitor functions and eases integration of third-party intrusion-detection tools 28, 30.

Each monitor 16 can analyze event records that form an event stream. The event stream may be derived from a variety of sources such as TCP/IP network packet contents or event records containing analysis reports disseminated by other monitors. For example, an event record can be formed from data included in the header and data segment of a network packet. The volume of packets transmitted and received, however, dictates careful assessment of ways to select and organize network packet information into event record streams.

Selection of packets can be based on different criteria. Streams of event records can be derived from discarded traffic (i.e., packets not allowed through the gateway because they violate filtering rules), pass-through traffic (i.e., packets allowed into the internal network from external sources), packets having a common protocol (e.g., all ICMP (Internet Control Message Protocol) packets that reach the gateway), packets involving network connection management (e.g., SYN, RESET, ACK, [window resize]), and packets targeting ports to which an administrator has not assigned any network service and that also remain unblocked by the firewall. Event streams may also be based on packet source addresses (e.g., packets whose source addresses match well-known external sites such as satellite offices or have raised suspicion from other monitoring efforts) or destination addresses (e.g., packets whose destination addresses match a given internal host or workstation). Selection can also implement application-layer monitoring (e.g., packets targeting a particular network service or application). Event records can also be produced from other sources of network packet information such as report logs produced by network entities. Event streams can be of very fine granularity. For example, a different stream might be derived for commands received from different commercial web-browsers since each web-browser produces different characteristic network activity.

A monitor 16 can also construct interval summary event records, which contain accumulated network traffic statistics (e.g., number of packets and number of kilobytes transferred). These event records are constructed at the end of each interval (e.g., once per N seconds). Event records are forwarded to the analysis engines 22, 24 for analysis.

The profile engine 22 can use a wide range of multivariate statistical measures to profile network activity indicated by an event stream. A statistical score represents how closely currently observed usage corresponds to the established patterns of usage. The profiler engine 22 separates profile management and the mathematical algorithms used to assess the anomaly of events. The profile engine 22 may use a statistical analysis technique described in A. Valdes and D. Anderson, "Statistical Methods for Computer Usage Anomaly Detection Using NIDES", Proceedings of the Third International Workshop on Rough Sets and Soft Computing, January 1995, which is incorporated by reference in its entirety. Such an engine 22 can profile network activity via one or more variables called measures. Measures can be categorized into four classes: categorical, continuous, intensity, and event distribution measures.

Categorical measures assume values from a discrete, nonordered set of possibilities. Examples of categorical measures include network source and destination addresses, commands (e.g., commands that control data transfer and manage network connections), protocols, error codes (e.g., privilege violations, malformed service requests, and malformed packet codes), and port identifiers. The profiler engine 22 can build empirical distributions of the category values encountered, even if the list of possible values is open-ended. The engine 22 can have mechanisms for "aging out" categories whose long-term probabilities drop below a threshold.

Continuous measures assume values from a continuous or ordinal set. Examples include inter-event time (e.g., difference in time stamps between consecutive events from the same stream), counting measures such as the number of errors of a particular type observed in the recent past, the volume of data transfers over a period of time, and network traffic measures (number of packets and number of kilobytes). The profiler engine 22 treats continuous measures by first allocating bins appropriate to the range of values of the underlying measure, and then tracking the frequency of observation of each value range. In this way, multi-modal distributions are accommodated and much of the computational machinery used for categorical measures is shared. Continuous measures are useful not only for intrusion detection, but also to support the monitoring of the health and status of the network from the perspective of connectivity and throughput. For example, a measure of traffic volume maintained can detect an abnormal loss in the data rate of received packets when this volume falls outside historical norms. This sudden drop can be specific both to the network entity being monitored and to the time of day (e.g., the average sustained traffic rate for a major network artery is much different at 11:00 a.m. than at midnight).

Intensity measures reflect the intensity of the event stream (e.g., number of ICMP packets) over specified time intervals (e.g., 1 minute, 10 minutes, and 1 hour). Intensity measures are particularly suited for detecting flooding attacks, while also providing insight into other anomalies.

Event distribution measures are meta-measures that describes how other measures in the profile are affected by each event. For example, an "ls" command in an FTP session affects the directory measure, but does not affect measures related to file transfer. This measure is not interesting for all event streams. For example, all network-traffic event records affect the same measures (number of packets and kilobytes) defined for that event stream, so the event distribution does not change. On the other hand, event distribution measures are useful in correlative analysis performed by a monitor 16*a*–16*f* that receives reports from other monitors 16*a*–16*f*.

The system maintains and updates a description of behavior with respect to these measure types in an updated profile. The profile is subdivided into short-term and long-term profiles. The short-term profile accumulates values between updates, and exponentially ages (e.g., weighs data based on how long ago the data was collected) values for comparison to the long-term profile. As a consequence of the aging mechanism, the short-term profile characterizes recent activity, where "recent" is determined by a dynamically configurable aging parameters. At update time (typically, a time of low system activity), the update function folds the short-term values observed since the last update into the long-term profile, and the short-term profile is cleared. The long-term profile is itself slowly aged to adapt to changes in subject activity. Anomaly scoring compares related attributes in the short-term profile against the long-term profile. As all evaluations are done against empirical distributions, no assumptions of parametric distributions are made, and multi-modal and categorical distributions are accommodated. Furthermore, the algorithms require no a priori knowledge of intrusive or exceptional activity.

The statistical algorithm adjusts a short-term profile for the measure values observed in the event record. The distribution of recently observed values is compared against the long-term profile, and a distance between the two is obtained. The difference is compared to a historically adaptive deviation. The empirical distribution of this deviation is transformed to obtain a score for the event. Anomalous events are those whose scores exceed a historically adaptive score threshold based on the empirical score distribution. This nonparametric approach handles all measure types and makes no assumptions on the modality of the distribution for continuous measures.

Profiles are provided to the computational engine as classes defined in the resource object 32. The mathematical functions for anomaly scoring, profile maintenance, and updating do not require knowledge of the data being analyzed beyond what is encoded in the profile class. Event collection interoperability supports translation of the event stream to the profile and measure classes. At that point, analysis for different types of monitored entities is mathematically similar. This approach imparts great flexibility to the analysis in that fading memory constants, update frequency, measure type, and so on are tailored to the network entity being monitored.

The measure types described above can be used individually or in combination to detect network packet attributes characteristic of intrusion. Such characteristics include large data transfers (e.g., moving or downloading files), an increase in errors (e.g., an increase in privilege violations or network packet rejections), network connection activity, and abnormal changes in network volume.

As shown, the monitor 16 also includes a signature engine 24. The signature engine 24 maps an event stream against abstract representations of event sequences that are known to indicate undesirable activity. Signature-analysis objectives depend on which layer in the hierarchical analysis scheme the signature engine operates. Service monitor 16*a*–16*c* signature engines 24 attempt to monitor for attempts to penetrate or interfere with the domain's operation. The signature engine scans the event stream for events that represent attempted exploitations of known attacks against the service, or other activity that stands alone as warranting a response from the monitor. Above the service layer, signature engines 24 scan the aggregate of intrusion reports from service monitors in an attempt to detect more global coordinated attack scenarios or scenarios that exploit interdependencies among network services. Layering signature engine analysis enables the engines 24 to avoid misguided searches along incorrect signature paths in addition to distributing the signature analysis.

A signature engines 24 can detect, for example, address spoofing, tunneling, source routing, SATAN attacks, and abuse of ICMP messages ("Redirect" and "Destination Unreachable" messages in particular). Threshold analysis is a rudimentary, inexpensive signature analysis technique that records the occurrence of specific events and, as the name implies, detects when the number of occurrences of that event surpasses a reasonable count. For example, monitors can encode thresholds to monitor activity such as the number of fingers, pings, or failed login requests to accounts such as guest, demo, visitor, anonymous FTP, or employees who have departed the company.

Signature engine 24 can also examine the data portion of packets in search of a variety of transactions that indicate suspicious, if not malicious, intentions by an external client. The signature engine 24, for example, can parse FTP traffic

traveling through the firewall or router for unwanted transfers of configuration or specific system data, or anonymous requests to access non-public portions of the directory structure. Similarly, a monitor can analyze anonymous FTP sessions to ensure that the file retrievals and uploads/modifications are limited to specific directories. Additionally, signature analysis capability can extend to session analyses of complex and dangerous, but highly useful, services like HTTP or Gopher.

Signature analysis can also scan traffic directed at unused ports (i.e., ports to which the administrator has not assigned a network service). Here, packet parsing can be used to study network traffic after some threshold volume of traffic, directed at an unused port, has been exceeded. A signature engine 24 can also employ a knowledge base of known telltale packets that are indicative of well-known network-service protocol traffic (e.g., FTP, Telnet, SMTP, HTTP). The signature engine 24 then determines whether the unknown port traffic matches any known packet sets. Such comparisons could lead to the discovery of network services that have been installed without an administrator's knowledge.

The analysis engines 22, 24 receive large volumes of events and produce smaller volumes of intrusion or suspicion reports that are then fed to the resolver 20. The resolver 20 is an expert system that receives the intrusion and suspicion reports produced by the analysis engines 22, 24 and reports produced externally by other analysis engines to which it subscribes. Based on these reports, the resolver 20 invokes responses. Because the volume of intrusion and suspicion reports is lower than the volume of events received by the analysis engines 22, 24, the resolver 20 can afford the more sophisticated demands of configuration maintenance and managing the response handling and external interfaces necessary for monitor operation. Furthermore, the resolver 20 adds to extensibility by providing the subscription interface through which third-party analysis tools 28, 30 can interact and participate in the hierarchical analysis scheme.

Upon its initialization, the resolver 20 initiates authentication and subscription sessions with those monitors 16*a*–16*f* whose identities appear in the monitor's 16 subscription-list (46 FIG. 3). The resolver 20 also handles all incoming requests by subscribers, which must authenticate themselves to the resolver 20. Once a subscription session is established with a subscriber monitor, the resolver 20 acts as the primary interface through which configuration requests are received and intrusion reports are disseminated.

Thus, resolvers 20 can request and receive reports from other resolvers at lower layers in the analysis hierarchy. The resolver 20 forwards analysis reports received from subscribees to the analysis engines 22, 24. This tiered collection and correlation of analysis results allows monitors 16*a*–16*f* to represent and profile global malicious or anomalous activity that is not visible locally.

In addition to external-interface responsibilities, the resolver 20 operates as a fully functional decision engine, capable of invoking real-time response measures in response to malicious or anomalous activity reports produced by the analysis engines. The resolver 20 also operates as the center of intramonitor communication. As the analysis engines 22, 24 build intrusion and suspicion reports, they propagate these reports to the resolver 20 for further correlation, response, and dissemination to other monitors 16*a*–16*f*. The resolver 20 can also submit runtime configuration requests to the analysis engines 22, 24, for example, to increase or decrease the scope of analyses (e.g., enable or disable additional signature rules) based on various operating metrics. These configuration requests could be made as a result of encountering other intrusion reports from other subscribers. For example, a report produced by a service monitor 16a–16c in one domain could be propagated to an enterprise monitor 16f, which in turn sensitizes service monitors in other domains to the same activity.

The resolver 20 also operates as the interface mechanism between administrators and the monitor 16. From the perspective of a resolver 20, the administrator interface is simply a subscribing service to which the resolver 20 may submit reports and receive configuration requests. An administrative interface tool can dynamically subscribe and unsubscribe to any of the deployed resolvers 20, as well as submit configuration requests and asynchronous probes as desired.

The monitors 16a–16f incorporate a bidirectional messaging system that uses a standard interface specification for communication within and between monitor elements and external modules. Using this interface specification, third-party modules 28, 30 can communicate with monitors. For example, third-party modules 28 can submit event records to the analysis engines 22, 24 for processing. Additionally, third-party modules 30 may also submit and receive analysis results via the resolver's 20 external interfaces. Thus, third-party modules 28, 30 can incorporate the results from monitors into other surveillance efforts or contribute their results to other monitors 16a–16f. Lastly, the monitor's 16 internal API allows third-party analysis engines to be linked directly into the monitor boundary.

The message system operates under an asynchronous communication model for handling results dissemination and processing that is generically referred to as subscription-based message passing. Component interoperation is client/server-based, where a client module may subscribe to receive event data or analysis results from servers. Once a subscription request is accepted by the server, the server module forwards events or analysis results to the client automatically as data becomes available, and may dynamically reconfigure itself as requested by the client's control requests. This asynchronous model reduces the need for client probes and acknowledgments.

The interface supports an implementation-neutral communication framework that separates the programmer's interface specification and the issues of message transport. The interface specification embodies no assumptions about implementation languages, host platform, or a network. The transport layer is architecturally isolated from the internals of the monitors so that transport modules may be readily introduced and replaced as protocols and security requirements are negotiated between module developers. The interface specification involves the definition of the messages that the various intrusion-detection modules must convey to one another and how these messages should be processed. The message structure and content are specified in a completely implementation-neutral context.

Both intramonitor and intermonitor communication employ identical subscription-based client-server models. With respect to intermonitor communication, the resolver 20 operates as a client to the analysis engines, and the analysis engines 22, 24 operate as clients to the event filters. Through the internal message system, the resolver 20 submits configuration requests to the analysis engines 22, 24, and receives from the analysis engines 22, 24 their analysis results. The analysis engines 22, 24 operate as servers providing the resolver 20 with intrusion or suspicion reports either asynchronously or upon request. Similarly, the analysis engines 22, 24 are responsible for establishing and maintaining a communication link with an event collection method (or event filter) and prompting the reconfiguration of the collection method's filtering semantics when necessary.

Intermonitor communication also operates using the subscription-based hierarchy. A domain monitor 16d–16e subscribes to the analysis results produced by service monitors 16a–16c, and then propagates its own analytical reports to its parent enterprise monitor 16f. The enterprise monitor 16f operates as a client to one or more domain monitors 16d–16e, allowing them to correlate and model enterprise-wide activity from the domain-layer results. Domain monitors 16d–16e operate as servers to the enterprise monitors 16f, and as clients to the service monitors 16a–16c deployed throughout their domain 12a–12c. This message scheme can operate substantially the same if correlation were to continue at higher layers of abstraction beyond enterprise 10 analysis.

Intramonitor and intermonitor programming interfaces are substantially the same. These interfaces can be subdivided into five categories of interoperation: channel initialization and termination, channel synchronization, dynamic configuration, server probing, and report/event dissemination. Clients are responsible for initiating and terminating channel sessions with servers. Clients are also responsible for managing channel synchronization in the event of errors in message sequencing or periods of failed or slow response (i.e., "I'm alive" confirmations). Clients may also submit dynamic configuration requests to servers. For example, an analysis engine 22, 24 may request an event collection method to modify its filtering semantics. Clients may also probe servers for report summaries or additional event information. Lastly, servers may send clients intrusion/suspicion reports in response to client probes or in an asynchronous dissemination mode.

The second part of the message system framework involves specification of a transport mechanism used to establish a given communication channel between monitors 16a–16f or possibly between a monitor 16a–16f and a third-party security module. All implementation dependencies within the message system framework are addressed by pluggable transport modules. Transport modules are specific to the participating intrusion-detection modules, their respective hosts, and potentially to the network—should the modules require cross-platform interoperation. Instantiating a monitor 16a–16f may involve incorporation of the necessary transport module(s) (for both internal and external communication).

The transport modules that handle intramonitor communication may be different from the transport modules that handle intermonitor communication. This allows the intramonitor transport modules to address security and reliability issues differently than how the intermonitor transport modules address security and reliability. While intramonitor communication may more commonly involve interprocess communication within a single host, intermonitor communication will most commonly involve cross-platform networked interoperation. For example, the intramonitor transport mechanisms may employ unnamed pipes which provides a kernel-enforced private interprocess communication channel between the monitor 16 components (this assumes a process hierarchy within the monitor 16 architecture). The monitor's 16 external transport, however, will more likely export data through untrusted network connections and thus require more extensive security management. To ensure the security and integrity of the message exchange, the external transport may employ public/private key authentication protocols and session key exchange. Using this same interface, third-party analysis tools may authenticate and exchange analysis results and configuration information in a well-defined, secure manner.

The pluggable transport permits flexibility in negotiating security features and protocol usage with third parties. Incorporation of a commercially available network management system can deliver monitoring results relating to security, reliability, availability, performance, and other attributes. The network management system may in turn subscribe to monitor produced results in order to influence network reconfiguration.

All monitors (service, domain, and enterprise) 16*a*–16*f* use the same monitor code-base. However, monitors may include different resource objects 32 having different configuration data and methods. This reusable software architecture can reduce implementation and maintenance efforts. Customizing and dynamically configuring a monitor 16 thus becomes a question of building and/or modifying the resource object 32.

Referring to FIG. 3, the resource object 32 contains the operating parameters for each of the monitor's 16 components as well as the analysis semantics (e.g., the profiler engine's 22 measure and category definition, or the signature engine's 24 penetration rule-base) necessary to process an event stream. After defining a resource object 32 to implement a particular set of analyses on an event stream, the resource object 32 may be reused by other monitors 16 deployed to analyze equivalent event streams. For example, the resource object 32 for a domain's router may be reused as other monitors 16 are deployed for other routers in a domain 12*a*–12*c*. A library of resource objects 32 provides prefabricated resource objects 32 for commonly available network entities.

The resource object 32 provides a pluggable configuration module for tuning the generic monitor code-base to a specific event stream. The resource object 32 includes configurable event structures 34, analysis unit configuration 38*a*–38*n*, engine configuration 40*a*–40*n*, resolver configuration 42, decision unit configuration 44, subscription list data 46, and response methods 48.

Configurable event structures 34 define the structure of event records and analysis result records. The monitor code-base maintains no internal dependence on the content or format of any given event stream or the analysis results produced from analyzing the event stream. Rather, the resource object 32 provides a universally applicable syntax for specifying the structure of event records and analysis results. Event records are defined based on the contents of an event stream(s). Analysis result structures are used to package the findings produced by analysis engines. Event records and analysis results are defined similarly to allow the eventual hierarchical processing of analysis results as event records by subscriber monitors.

Event-collection methods 36 gather and parse event records for analysis engine processing. Processing by analysis engines is controlled by engine configuration 40*a*–40*n* variables and data structures that specify the operating configuration of a fielded monitor's analysis engine(s). The resource object 32 maintains a separate collection of operating parameters for each analysis engine instantiated in the monitor 16. Analysis unit configuration 38*a*–38*n* include configuration variables that define the semantics employed by the analysis engine to process the event stream.

The resolver configuration 42 includes operating parameters that specify the configuration of the resolver's internal

modules. The decision unit configuration 44 describes semantics used by the resolver's decision unit for merging the analysis results from the various analysis engines. The semantics include the response criteria used to invoke countermeasure handlers. A resource object 32 may also include response methods 48. Response methods 48 include preprogrammed countermeasure methods that the resolver may invoke as event records are received. A response method 48 includes evaluation metrics for determining the circumstances under which the method should be invoked. These metrics include a threshold metric that corresponds to the measure values and scores produced by the profiler engine 22 and severity metrics that correspond to subsets of the associated attack sequences defined within the resource object 32.

Countermeasures range from very passive responses, such as report dissemination to other monitors 16*a*–16*f* or administrators, to highly aggressive actions, such as severing a communication channel or the reconfiguration of logging facilities within network components (e.g., routers, firewalls, network services, audit daemons). An active response may invoke handlers that validate the integrity of network services or other assets to ensure that privileged network services have not been subverted. Monitors 16*a*–16*f* may invoke probes in an attempt to gather as much counterintelligence about the source of suspicious traffic by using features such as traceroute or finger.

The resource object 32 may include a subscription list 46 that includes information necessary for establishing subscription-based communication sessions, which may include network address information and public keys used by the monitor to authenticate potential clients and servers. The subscription list 46 enables transmission or reception of messages that report malicious or anomalous activity between monitors. The most obvious examples where relationships are important involve interdependencies among network services that make local policy decisions. For example, the interdependencies between access checks performed during network file system mounting and the IP mapping of the DNS service. An unexpected mount monitored by the network file system service may be responded to differently if the DNS monitor informs the network file system monitor of suspicious updates to the mount requestor's DNS mapping.

The contents of the resource object 32 are defined and utilized during monitor 16 initialization. In addition, these fields may be modified by internal monitor 16 components, and by authorized external clients using the monitor's 16 API. Modifying the resource object 32 permits adaptive analysis of an event stream, however, it also introduces a potential stability problem if dynamic modifications are not tightly restricted to avoid cyclic modifications. To address this issue, monitors 16 can be configured to accept configuration requests from only higher-level monitors 16.

Referring to FIG. 4, a monitor performs network surveillance by monitoring 66 a stream of network packets. The monitor builds a statistical model of network activity from the network packets, for example, by building 68 long-term and short-term statistical profiles from measures derived from the network packets. The measures include measures that can show anomalous network activity characteristic of network intrusion such as measures that describe data transfers, network connections, privilege and network errors, and abnormal levels of network traffic. The monitor can compare 70 the long-term and short-term profiles to detect suspicious network activity. Based on this comparison, the monitor can respond 72 by reporting the activity to another monitor or by executing a countermeasure response. More information can be found in P. Porras and A. Valdes "Live Traffic Analysis of TCP/IP Gateways", Networks and Distributed Systems Security Symposium, March 1998, which is incorporated by reference in its entirety.

A few examples can illustrate this method of network surveillance. Network intrusion frequently causes large data transfers, for example, when an intruder seeks to download sensitive files or replace system files with harmful substitutes. A statistical profile to detect anomalous data transfers might include a continuous measure of file transfer size, a categorical measure of the source or destination directory of the data transfer, and an intensity measure of commands corresponding to data transfers (e.g., commands that download data). These measures can detect a wide variety of data transfer techniques such as a large volume of small data transfers via e-mail or downloading large files en masse. The monitor may distinguish between network packets based on the time such packets were received by the network entity, permitting statistical analysis to distinguish between a normal data transfer during a workday and an abnormal data transfer on a weekend evening.

Attempted network intrusion may also produce anomalous levels of errors. For example, categorical and intensity measures derived from privilege errors may indicate attempts to access protected files, directories, or other network assets. Of course, privilege errors occur during normal network operation as users mistype commands or attempt to perform an operation unknowingly prohibited. By comparing the long-term and short-term statistical profiles, a monitor can distinguish between normal error levels and levels indicative of intrusion without burdening a network administrator with the task of arbitrarily setting an unvarying threshold. Other measures based on errors, such as codes describing why a network entity rejected a network packet enable a monitor to detect attempts to infiltrate a network with suspicious packets.

Attempted network intrusion can also be detected by measures derived from network connection information. For example, a measure may be formed from the correlation (e.g., a ratio or a difference) of the number of SYN connection request messages with the number of SYN_ACK connection acknowledgment messages and/or the number of ICMP messages sent. Generally, SYN requests received should balance with respect to the total of SYN_ACK and ICMP messages sent. That is, flow into and out-of a network entity should be conserved. An imbalance can indicate repeated unsuccessful attempts to connect with a system, perhaps corresponding to a methodical search for an entry point to a system. Alternatively, intensity measures of transport-layer connection requests, such as a volume analysis of SYN-RST messages, could indicate the occurrence of a SYN-attack against port availability or possibly port-scanning. Variants of this can include intensity measures of TCP/FIN messages, considered a more stealthy form of port scanning.

Many other measures can detect network intrusion. For example, "doorknob rattling," testing a variety of potentially valid commands to gain access (e.g., trying to access a "system" account with a password of "system"), can be detected by a variety of categorical measures. A categorical measure of commands included in network packets can identify an unusual short-term set of commands indicative of "doorknob-rattling." Similarly, a categorical measure of protocol requests may also detect an unlikely mix of such requests.

Measures of network packet volume can also help detect malicious traffic, such as traffic intended to cause service denials or perform intelligence gathering, where such traffic may not necessarily be violating filtering policies. A measure reflecting a sharp increase in the overall volume of discarded packets as well as a measure analyzing the disposition of the discarded packets can provide insight into unintentionally malformed packets resulting from poor line quality or internal errors in neighboring hosts. High volumes of discarded packets can also indicate more maliciously intended transmissions such as scanning of UPD ports or IP address scanning via ICMP echoes. Excessive number of mail expansion request commands (EXPN) may indicate intelligence gathering, for example, by spammers.

A long-term and short-term statistical profile can be generated for each event stream. Thus, different event streams can "slice" network packet data in different ways. For example, an event stream may select only network packets having a source address corresponding to a satellite office. Thus, a long-term and short-term profile will be generated for the particular satellite office. Thus, although a satellite office may have more privileges and should be expected to use more system resources than other external addresses, a profile of satellite office use can detect "address spoofing" (i.e., modifying packet information to have a source address of the satellite office).

The same network packet event may produce records in more than one event stream. For example, one event stream may monitor packets for FTP commands while another event stream monitors packets from a particular address. In this case, an FTP command from the address would produce an event record in each stream.

Referring to FIG. 5, a monitor may also "deinterleave." That is, the monitor may create and update 74, 76 more than one short-term profile for comparison 78 against a single long-term profile by identifying one of the multiple short-term profiles that will be updated by an event record in an event stream. For example, at any one time a network entity may handle several FTP "anonymous" sessions. If each network packet for all anonymous sessions were placed in a single short-term statistical profile, potentially intrusive activity of one anonymous session may be statistically ameliorated by non-intrusive sessions. By creating and updating short-term statistical profiles for each anonymous session, each anonymous session can be compared against the long-term profile of a normal FTP anonymous session. Deinterleaving can be done for a variety of sessions including HTTP sessions (e.g., a short-term profile for each browser session).

Referring to FIG. 6, a computer platform 14 suitable for executing a network monitor 16 includes a display 50, a keyboard 54, a pointing device 58 such as a mouse, and a digital computer 56. The digital computer 56 includes memory 62, a processor 60, a mass storage device 64*a*, and other customary components such as a memory bus and peripheral bus. The platform 14 may further include a network connection 52.

Mass storage device 64*a* can store instructions that form a monitor 16. The instructions may be transferred to memory 62 and processor 60 in the course of operation. The instructions 16 can cause the display 50 to display images via an interface such as a graphical user interface. Of course, instructions may be stored on a variety of mass storage devices such as a floppy disk 64*b*, CD-ROM 64*c*, or PROM (not shown).

Other embodiments are within the scope of the following claims.

What is claimed is:

1. A computer-automated method of hierarchical event monitoring and analysis within an enterprise network comprising:

deploying a plurality of network monitors in the enterprise network;

detecting, by the network monitors, suspicious network activity based on analysis of network traffic data selected from one or more of the following categories: {network packet data transfer commands, network packet data transfer errors, network packet data volume, network connection requests, network connection denials, error codes included in a network packet, network connection acknowledgements, and network packets indicative of well-known network-service protocols};

generating, by the monitors, reports of said suspicious activity; and

automatically receiving and integrating the reports of suspicious activity, by one or more hierarchical monitors.

2. The method of claim 1, wherein integrating comprises correlating intrusion reports reflecting underlying commonalities.

3. The method of claim 1, wherein integrating further comprises invoking countermeasures to a suspected attack.

4. The method of claim 1, wherein the plurality of network monitors include an API for encapsulation of monitor functions and integration of third-party tools.

5. The method of claim 1, wherein the enterprise network is a TCP/IP network.

6. The method of claim 1, wherein the network monitors are deployed at one or more of the following facilities of the enterprise network: {gateways, routers, proxy servers}.

7. The method of claim 1, wherein at least one of said network monitors utilizes a statistical detection method.

8. The method of claim 1, wherein deploying the network monitors includes placing a plurality of service monitors among multiple domains of the enterprise network.

9. The method of claim 8, wherein receiving and integrating is performed by a domain monitor with respect to a plurality of service monitors within the domain monitor's associated network domain.

10. The method of claim 1, wherein deploying the network monitors includes deploying a plurality of domain monitors within the enterprise network, each domain monitor being associated with a corresponding domain of the enterprise network.

11. The method of claim 10, wherein receiving and integrating is performed by an enterprise monitor with respect to a plurality of domain monitors within the enterprise network.

12. The method of claim 10, wherein the plurality of domain monitors within the enterprise network establish peer-to-peer relationships with one another.

13. An enterprise network monitoring system comprising:

a plurality of network monitors deployed within an enterprise network, said plurality of network monitors detecting suspicious network activity based on analysis of network traffic data selected from one or more of the following categories: {network packet data transfer commands, network packet data transfer errors, network packet data volume, network connection requests, network connection denials, error codes included in a network packet, network connection acknowledgements, and network packets indicative of well-known network-service protocols};

said network monitors generating reports of said suspicious activity; and

one or more hierarchical monitors in the enterprise network, the hierarchical monitors adapted to automatically receive and integrate the reports of suspicious activity.

14. The system of claim 13, wherein the integration comprises correlating intrusion reports reflecting underlying commonalities.

15. The system of claim 13, wherein the integration further comprises invoking countermeasures to a suspected attack.

16. The system of claim 13, wherein the plurality of network monitors include an application programming interface (API) for encapsulation of monitor functions and integration of third-party tools.

17. The system of claim 13, wherein the enterprise network is a TCP/IP network.

18. The system of claim 13, wherein the network monitors are deployed at one or more of the following facilities of the enterprise network: {gateways, routers, proxy servers}.

19. The system of claim 13, wherein the plurality of network monitors includes a plurality of service monitors among multiple domains of the enterprise network.

20. The system of claim 19, wherein a domain monitor associated with the plurality of service monitors within the domain monitor's associated network domain is adapted to automatically receive and integrate the reports of suspicious activity.

21. The system of claim 13, wherein the plurality of network monitors include a plurality of domain monitors within the enterprise network, each domain monitor being associated with a corresponding domain of the enterprise network.

22. The system of claim 21, wherein an enterprise monitor associated with a plurality of domain monitors is adapted to automatically receive and integrate the reports of suspicious activity.

23. The system of claim 21, wherein the plurality of domain monitors within the enterprise network interface as a plurality of peer-to-peer relationships with one another.

24. A computer-automated method of hierarchical event monitoring and analysis within an enterprise network comprising:

deploying a plurality of network monitors in the enterprise network, wherein the enterprise network is a virtual private network (VPN);

detecting, by the network monitors, suspicious network activity based on analysis of network traffic data;

generating, by the monitors, reports of said suspicious activity; and

automatically receiving and integrating the reports of suspicious activity, by one or more hierarchical monitors.

25. The method of claim 24, wherein said integrating comprises correlating intrusion reports reflecting underlying commonalities.

26. The method of claim 24, wherein said integrating further comprises invoking countermeasures to a suspected attack.

27. The method of claim 24, wherein the plurality of network monitors include an API for encapsulation of monitor functions and integration of third-party tools.

28. The method of claim 24, wherein said network traffic data is selected from one or more of the following categories: {network packet data transfer commands, network

packet data transfer errors, network packet data volume, network connection requests, network connection denials, error codes included in a network packet}.

29. The method of claim 24, wherein said deploying the network monitors includes placing a plurality of service monitors among multiple domains of the enterprise network.

30. The method of claim 29, wherein said receiving and integrating is performed by a domain monitor with respect to a plurality of service monitors within the domain monitor's associated network domain.

31. The method of claim 24, wherein said deploying the network monitors includes deploying a plurality of domain monitors within the enterprise network, each domain monitor being associated with a corresponding domain of the enterprise network.

32. The method of claim 31, wherein said receiving and integrating is performed by an enterprise monitor with respect to a plurality of domain monitors within the enterprise network.

33. The method of claim 31, wherein the plurality of domain monitors within the enterprise network establish peer-to-peer relationships with one another.

34. A computer-automated method of hierarchical event monitoring and analysis within an enterprise network comprising:

- deploying a plurality of network monitors in the enterprise network, wherein at least one of the network monitors is deployed at a gateway;
- detecting, by the network monitors, suspicious network activity based on analysis of network traffic data;
- generating, by the monitors, reports of said suspicious activity; and
- automatically receiving and integrating the reports of suspicious activity, by one or more hierarchical monitors.

35. The method of claim 34, wherein said integrating comprises correlating intrusion reports reflecting underlying commonalities.

36. The method of claim 34, wherein said integrating further comprises invoking countermeasures to a suspected attack.

37. The method of claim 34, wherein the plurality of network monitors include an API for encapsulation of monitor functions and integration of third-party tools.

38. The method of claim 34, wherein said network traffic data is selected from one or more of the following categories: {network packet data transfer commands, network packet data transfer errors, network packet data volume, network connection requests, network connection denials, error codes included in a network packet}.

39. The method of claim 34, wherein said deploying the network monitors includes placing a plurality of service monitors among multiple domains of the enterprise network.

40. The method of claim 39, wherein said receiving and integrating is performed by a domain monitor with respect to a plurality of service monitors within the domain monitor's associated network domain.

41. The method of claim 34, wherein said deploying the network monitors includes deploying a plurality of domain monitors within the enterprise network, each domain monitor being associated with a corresponding domain of the enterprise network.

42. The method of claim 41, wherein said receiving and integrating is performed by an enterprise monitor with respect to a plurality of domain monitors within the enterprise network.

43. The method of claim 41, wherein the plurality of domain monitors within the enterprise network establish peer-to-peer relationships with one another.

44. A computer-automated method of hierarchical event monitoring and analysis within an enterprise network comprising:

- deploying a plurality of network monitors in the enterprise network, wherein at least one of the network monitors is deployed at a router;
- detecting, by the network monitors, suspicious network activity based on analysis of network traffic data;
- generating, by the monitors, reports of said suspicious activity; and
- automatically receiving and integrating the reports of suspicious activity, by one or more hierarchical monitors.

45. The method of claim 44, wherein said integrating comprises correlating intrusion reports reflecting underlying commonalities.

46. The method of claim 44, wherein said integrating further comprises invoking countermeasures to a suspected attack.

47. The method of claim 44, wherein the plurality of network monitors include an API for encapsulation of monitor functions and integration of third-party tools.

48. The method of claim 44, wherein said network traffic data is selected from one or more of the following categories: {network packet data transfer commands, network packet data transfer errors, network packet data volume, network connection requests, network connection denials, error codes included in a network packet}.

49. The method of claim 44, wherein said deploying the network monitors includes placing a plurality of service monitors among multiple domains of the enterprise network.

50. The method of claim 49, wherein said receiving and integrating is performed by a domain monitor with respect to a plurality of service monitors within the domain monitor's associated network domain.

51. The method of claim 44, wherein said deploying the network monitors includes deploying a plurality of domain monitors within the enterprise network, each domain monitor being associated with a corresponding domain of the enterprise network.

52. The method of claim 51, wherein said receiving and integrating is performed by an enterprise monitor with respect to a plurality of domain monitors within the enterprise network.

53. The method of claim 51, wherein the plurality of domain monitors within the enterprise network establish peer-to-peer relationships with one another.

54. A computer-automated method of hierarchical event monitoring and analysis within an enterprise network comprising:

- deploying a plurality of network monitors in the enterprise network, wherein at least one of the network monitors is deployed at a proxy server;
- detecting, by the network monitors, suspicious network activity based on analysis of network traffic data;
- generating, by the monitors, reports of said suspicious activity; and
- automatically receiving and integrating the reports of suspicious activity, by one or more hierarchical monitors.

55. The method of claim 54, wherein said integrating comprises correlating intrusion reports reflecting underlying commonalities.

56. The method of claim 54, wherein said integrating further comprises invoking countermeasures to a suspected attack.

57. The method of claim 54, wherein the plurality of network monitors include an API for encapsulation of monitor functions and integration of third-party tools.

58. The method of claim 54, wherein said network traffic data is selected from one or more of the following categories: {network packet data transfer commands, network packet data transfer errors, network packet data volume, network connection requests, network connection denials, error codes included in a network packet}.

59. The method of claim 54, wherein said deploying the network monitors includes placing a plurality of service monitors among multiple domains of the enterprise network.

60. The method of claim 59, wherein said receiving and integrating is performed by a domain monitor with respect to a plurality of service monitors within the domain monitor's associated network domain.

61. The method of claim 54, wherein said deploying the network monitors includes deploying a plurality of domain monitors within the enterprise network, each domain monitor being associated with a corresponding domain of the enterprise network.

62. The method of claim 61, wherein said receiving and integrating is performed by an enterprise monitor with respect to a plurality of domain monitors within the enterprise network.

63. The method of claim 61, wherein the plurality of domain monitors within the enterprise network establish peer-to-peer relationships with one another.

64. A computer-automated method of hierarchical event monitoring and analysis within an enterprise network comprising:

deploying a plurality of network monitors in the enterprise network, wherein at least one of the network monitors is deployed at a firewall;

detecting, by the network monitors, suspicious network activity based on analysis of network traffic data;

generating, by the monitors, reports of said suspicious activity; and

automatically receiving and integrating the reports of suspicious activity, by one or more hierarchical monitors.

65. The method of claim 64, wherein said integrating comprises correlating intrusion reports reflecting underlying commonalities.

66. The method of claim 64, wherein said integrating further comprises invoking countermeasures to a suspected attack.

67. The method of claim 64, wherein the plurality of network monitors include an API for encapsulation of monitor functions and integration of third-party tools.

68. The method of claim 64, wherein said network traffic data is selected from one or more of the following categories: {network packet data transfer commands, network packet data transfer errors, network packet data volume, network connection requests, network connection denials, error codes included in a network packet}.

69. The method of claim 64, wherein said deploying the network monitors includes placing a plurality of service monitors among multiple domains of the enterprise network.

70. The method of claim 69, wherein said receiving and integrating is performed by a domain monitor with respect to a plurality of service monitors within the domain monitor's associated network domain.

71. The method of claim 64, wherein said deploying the network monitors includes deploying a plurality of domain monitors within the enterprise network, each domain monitor being associated with a corresponding domain of the enterprise network.

72. The method of claim 71, wherein said receiving and integrating is performed by an enterprise monitor with respect to a plurality of domain monitors within the enterprise network.

73. The method of claim 71, wherein the plurality of domain monitors within the enterprise network establish peer-to-peer relationships with one another.

74. An enterprise network monitoring system comprising:

a plurality of network monitors deployed within an enterprise network, wherein the enterprise network is a virtual private network (VPN), said plurality of network monitors detecting suspicious network activity based on analysis of network traffic data;

said network monitors generating reports of said suspicious activity; and

one or more hierarchical monitors in the enterprise network, the hierarchical monitors adapted to automatically receive and integrate the reports of suspicious activity.

75. The system of claim 74, wherein the integration comprises correlating intrusion reports reflecting underlying commonalities.

76. The system of claim 74, wherein the integration further comprises invoking countermeasures to a suspected attack.

77. The system of claim 74, wherein the plurality of network monitors include an application programming interface (API) for encapsulation of monitor functions and integration of third-party tools.

78. The system of claim 74, wherein said network traffic data is selected from one or more of the following categories: {network packet data transfer commands, network packet data transfer errors, network packet data volume, network connection requests, network connection denials, error codes included in a network packet}.

79. The system of claim 74, wherein the plurality of network monitors includes a plurality of service monitors among multiple domains of the enterprise network.

80. The system of claim 79, wherein a domain monitor associated with the plurality of service monitors within the domain monitor's associated network domain is adapted to automatically receive and integrate the reports of suspicious activity.

81. The system of claim 74, wherein the plurality of network monitors include a plurality of domain monitors within the enterprise network, each domain monitor being associated with a corresponding domain of the enterprise network.

82. The system of claim 81, wherein an enterprise monitor associated with a plurality of domain monitors is adapted to automatically receive and integrate the reports of suspicious activity.

83. The system of claim 81, wherein the plurality of domain monitors within the enterprise network interface as a plurality of peer-to-peer relationships with one another.

84. An enterprise network monitoring system comprising:

a plurality of network monitors deployed within an enterprise network, wherein at least one of the network monitors is deployed at one or more of the following facilities of the enterprise network: {gateways, routers, proxy servers, firewalls}, said plurality of network monitors detecting suspicious network activity based on analysis of network traffic data;

said network monitors generating reports of said suspicious activity; and

one or more hierarchical monitors in the enterprise network, the hierarchical monitors adapted to automatically receive and integrate the reports of suspicious activity.

85. The system of claim 84, wherein the integration comprises correlating intrusion reports reflecting underlying commonalities.

86. The system of claim 84, wherein the integration further comprises invoking countermeasures to a suspected attack.

87. The system of claim 84, wherein the plurality of network monitors include an application programming interface (API) for encapsulation of monitor functions and integration of third-party tools.

88. The system of claim 84, wherein said network traffic data is selected from one or more of the following categories: {network packet data transfer commands, network packet data transfer errors, network packet data volume, network connection requests, network connection denials, error codes included in a network packet}.

89. The system of claim 84, wherein the plurality of network monitors includes a plurality of service monitors among multiple domains of the enterprise network.

90. The system of claim 89, wherein a domain monitor associated with the plurality of service monitors within the domain monitor's associated network domain is adapted to automatically receive and integrate the reports of suspicious activity.

91. The system of claim 84, wherein the plurality of network monitors include a plurality of domain monitors within the enterprise network, each domain monitor being associated with a corresponding domain of the enterprise network.

92. The system of claim 91, wherein an enterprise monitor associated with a plurality of domain monitors is adapted to automatically receive and integrate the reports of suspicious activity.

93. The system of claim 91, wherein the plurality of domain monitors within the enterprise network interface as a plurality of peer-to-peer relationships with one another.

* * * * *

(12) **United States Patent**
**Porras et al.**

(10) **Patent No.: US 6,708,212 B2**
(45) **Date of Patent: Mar. 16, 2004**

(54) **NETWORK SURVEILLANCE**

(75) Inventors: **Phillip Andrew Porras**, Cupertino, CA (US); **Alfonso Valdes**, San Carlos, CA (US)

(73) Assignee: **SRI International**, Menlo Park, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **10/429,607**

(22) Filed: **May 5, 2003**

(65) **Prior Publication Data**

US 2003/0212903 A1 Nov. 13, 2003

**Related U.S. Application Data**

(63) Continuation of application No. 10/254,457, filed on Sep. 25, 2002, which is a continuation of application No. 09/658,137, filed on Sep. 8, 2000, now Pat. No. 6,484,203, which is a continuation of application No. 09/188,739, filed on Nov. 9, 1998, now Pat. No. 6,321,338.

(51) **Int. Cl.**$^7$ .......................... **G06F 11/30**; G06F 12/14
(52) **U.S. Cl.** .......................................... **709/224**; 713/201
(58) **Field of Search** ................................ 709/223–225; 713/200, 201

(56) **References Cited**

U.S. PATENT DOCUMENTS

| | | | | |
|---|---|---|---|---|
| 4,672,609 | A | 6/1987 | Humphrey et al. | 371/21 |
| 4,773,028 | A | 9/1988 | Tallman | 364/550 |
| 5,210,704 | A | 5/1993 | Husseiny | 364/551.01 |
| 5,440,723 | A | 8/1995 | Arnold et al. | 395/181 |
| 5,539,659 | A | 7/1996 | McKee et al. | 709/224 |
| 5,557,742 | A | 9/1996 | Smaha et al. | 395/186 |
| 5,706,210 | A | 1/1998 | Kumano et al. | 709/224 |
| 5,748,098 | A | 5/1998 | Grace | 340/825.16 |
| 5,790,799 | A | 8/1998 | Mogul | 709/224 |
| 5,878,420 | A | 3/1999 | De la Salle | 707/10 |
| 5,919,258 | A | 7/1999 | Kayashima et al. | 713/201 |
| 5,922,051 | A | 7/1999 | Sidey | 709/223 |
| 5,940,591 | A | 8/1999 | Boyle et al. | 395/187.01 |
| 5,974,237 | A | 10/1999 | Shurmer et al. | 709/224 |
| 5,974,457 | A | 10/1999 | Waclawsky et al. | 709/224 |
| 5,991,881 | A | 11/1999 | Conklin et al. | 713/201 |
| 6,009,467 | A | 12/1999 | Ratcliff et al. | 709/224 |
| 6,052,709 | A | 4/2000 | Paul | 709/202 |
| 6,070,244 | A | 5/2000 | Orchier et al. | 713/201 |
| 6,144,961 | A | 11/2000 | de la Salle | 707/10 |

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

| | | | |
|---|---|---|---|
| WO | 99/13427 | 3/1999 | G06K/7/00 |
| WO | 99/57626 | 11/1999 | G06F/1/16 |
| WO | 00/10278 | 2/2000 | |
| WO | 00/25214 | 5/2000 | G06F/12/14 |
| WO | 00/25527 | 5/2000 | H04Q/3/00 |
| WO | 00/34867 | 6/2000 | G06F/11/30 |
| WO | 02/101516 | 12/2002 | |

OTHER PUBLICATIONS

Hartley, B., "Intrusion Detection Systems: What You Need to Know," Business Security Advisor Magazine, Doc#05257, allegedly dated Sep. 1998, http://advisor.com/doc/05257, 7 pages, printed Jun. 10, 2003.

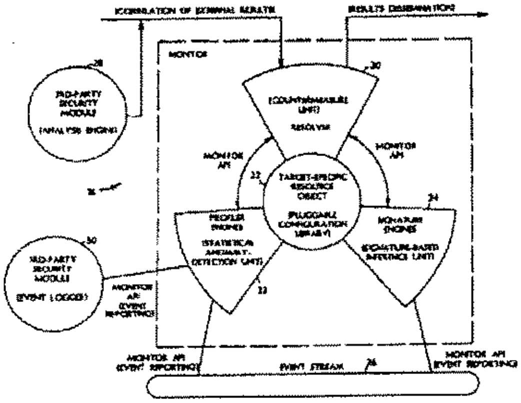
(List continued on next page.)

*Primary Examiner*—Thomas M. Heckler
(74) *Attorney, Agent, or Firm*—Kin-Wah Tong; Moser, Patterson & Sheridan, LLP.

(57) **ABSTRACT**

A method of network surveillance includes receiving network packets handled by a network entity and building at least one long-term and a least one short-term statistical profile from a measure of the network packets that monitors data transfers, errors, or network connections. A comparison of the statistical profiles is used to determine whether the difference between the statistical profiles indicates suspicious network activity.

**24 Claims, 5 Drawing Sheets**



U.S. PATENT DOCUMENTS

| | | | | |
|---|---|---|---|---|
| 6,396,845 | B1 | 5/2002 | Sugita | 709/224 X |
| 6,453,346 | B1 | 9/2002 | Garg et al. | 709/224 |
| 6,460,141 | B1 | 10/2002 | Olden | 712/201 |
| 6,519,703 | B1 | 2/2003 | Joyce | 713/201 |
| 2002/0032717 | A1 | 3/2002 | Malan et al. | 709/105 |
| 2002/0032793 | A1 | 3/2002 | Malan et al. | 709/232 |
| 2002/0032880 | A1 | 3/2002 | Poletto et al. | 714/4 |
| 2002/0035698 | A1 | 3/2002 | Malan et al. | 713/201 |
| 2002/0138753 | A1 | 9/2002 | Munson | 713/200 |
| 2002/0144156 | A1 | 10/2002 | Copeland, III | 713/201 |
| 2003/0037136 | A1 | 2/2003 | Labovitz et al. | 709/224 |

OTHER PUBLICATIONS

Hurwicz, M., "Cracker Tracking: Tighter Security with Intrusion Detection," BYTE.com, allegedly dated May 1998, www.byte.com/art/9805/sec20/art1.htm, 8 pages, printed Jun. 10, 2003.

"Networkers, Intrusion Detection and Scanning with Active Audit," Session 1305, ©1998Cisco Systems, www.cisco.com/networkers/nw99_pres/1305.pdf, 0893–04F9_c3.scr, printed Jun. 10, 2003.

Paller, A., "About the Shadow Intrusion Detection System" Linux Weekly News, allegedly dated Sep. 1998, lwn.net/1998/0910/shadowhtml, 38 pages, printed Jun. 10, 2003.

Cisco Secure Intrusion Detection System, Release 2.1.1, NetRanger User's Guide, Version 2.1.1, ©1998, Cisco Systems, Inc., allegedly released on Apr. 1998, www.cisco.com/univercd/cc/td/doc/product/iaabu/csids/csids3/index.htm, printed Jun. 10, 2003, 334 pages, (See CSI document listed at C7 below).

Cisco Secure Intrusion Detection System 2.1.1 Release Notes, Table of Contents, Release Notes for NetRanger 2.1.1, ©1992–2002, Cisco Systems, Inc., , allegedly posted Sep. 28, 2002, 29 pages, www.cisco.com/univercd/cc/td/doc/product/iaabu/csids/csids3/nr11new.htm, printed Jun. 10, 2003.

R. Power, et al., "CSI Intrusion Detection System Resource", allegedly dated Jul. 1998, 216.239.57.100/search?q=cache:gvTCojxD6nMJ:www.gocsi.com/ques.htm+site:www.gocsi.com+ques&hl=en&ie=UTF–8, printed Jun. 16, 2003.

Debar, et al., "Towards a Taxonomy of Intrusion–Detection Systems," Computer Networks 31 (1999), 805–822.

Debar et al., "A Neural Network Component for an Intrusion Detection System," © 1992 IEEE.

Denning et al, "Prototype IDES: A Real–Time Intrusion–Detection Expert System," SRI Project ECU 7508, SRI International, Menlo Park, California, Aug. 1987.

Denning et al., "Requirements and Model for IDES—a Real–Time Intrusion–Detection Expert System," SRI Project 6169, SRI International, Menlo Park, CA, Aug. 1985.

Denning, "An Intrusion–Detection Model," SRI International, Menlo Park, CA Technical Report CSL–149, Nov. 1985.

Dowell, "The Computerwatch Data Reduction Tool," AT&T Bell Laboratories, Whippany, New Jersey.

Fox, et al., "A Neural Network Approach Towards Intrusion Detection," Harris Corporation, Government Information Systems Division, Melbourne, FL, Jul. 2, 1990.

Garvey, et al., "Model–Based Intrusion Detection," Proceedings of the 14th national Computer Security Conference, Washington, DC, Oct. 1991.

Garvey, et al., "An Inference Technique for Integrating Knowledge from Disparate Sources," Proc. IJCAI, Vancouver, BC, Aug. 1981, 319–325.

Ilgun et al., State Transition Analysis: A Rule–Based Intrusion Detection Approach, IEEE Transactions on Software Engineering, vol., 21, No. 3, Mar. 1995.

Javitz et al., "The SRI IDES Statistical Anomaly Detector," Proceedings, 1991 IEEE Symposium on Security and Privacy, Oakland, California, May 1991.

Jarvis et al., The NIDES Statistical Component Description and Justification, SRI International Annual Report A010, Mar. 7, 1994.

Kaven, "The Digital Dorman," PC Magazine, Nov. 16, 1999.

Liepins, et al., "Anomaly Detection; Purpose and Framework," US DOE Office of Safeguards and Security.

Lindquist, et al., "Detecting Computer and Network Misuse Through the Production–Based Expert System Toolset (P–BEST)," Oct. 25, 1998.

Lunt et al., "An Expert System to Classify and Sanitize Text," SRI International, Computer Science Laboratory, Menlo Park, CA.

Lunt, "A Survey of Intrusion Detection Techniques," Computers & Security, 12 (1993) 405–418.

Lunt, "Automated Audit Trail Analysis and Intrusion Detection: A Survey," Proceedings of the 11th National Computer Security Conference, Baltimore, MD, Oct. 1988.

Lunt, et al., "Knowledge–Based Intrusion Detection Expert System," Proceedings of the 1988 IEE Symposium on Security and Privacy, Apr. 1988.

Porras et al, "Emerald: Event Monitoring Enabling Responses to Anomalous Live Disturbances," 20th NISSC—Oct. 9, 1997.

Porras et al., Penetration State Transition Analysis A Rule–Based Intrusion Detection Approach, © 1992 IEEE.

Sebring et al., Expert Systems in Intrusion Detection: A Case Study.

Shieh et al., A Pattern–Oriented Intrusion–Detection Model and Its Applications © 1991 IEEE.

Smaha, Haystack: An Intrusion Detection System: © 1988 IEEE Computer Society Press: Proceedings of the Fourth Aerospace Computer Security Application Conference, 1988, pp. 37–44.

Snapp, Signature Analysis and Communication Issues in a Distributed Intrusion Detection System,: Thesis 1991.

Snapp et al., "DIDS (Distributed Intrusion Detection System)—Motivation, Architecture and An Early Prototype," Computer Security Laboratory, Division of Computer Science, Unic. Of California, Davis, Davis, CA.

Tener, "AI & 4GL: Automated Detection and Investigation Tools," Computer Security in the Age of Information, Proceedings of the Fifth IFIP International Conference on Computer Security, W.J. Caelli (ed.).

Teng et al., "Adaptive Real–Time Anomaly Detection Using Inductively Generated Sequential Patterns," © 1990.

Vaccaro et al., "Detection of Anomalous Computer Session Activity," © 1989 IEEE

Weiss, "Analysis of Audit and Protocol Data using Methods from Artificial Intelligence," Siemens, AG, Munich, West Germany.

Winkler, "A UNIX Prototype for Intrusion and Anomaly Detection in Secure Networks," © Planning Research Corp. 1990.

Lunt et al., "A Prototype Real–Time Intrusion–Detection Expert System," Proceedings of the 1988 IEEE Symposium on Security and Privacy, Apr. 1988.

Boyen, et al., "Tractable Inference for Complex Stochastic Processes," Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence (UAI–98), pp. 33–42, Madison, WI, Jul. 24–26, 1998.

Copeland, J., "Observing Network Traffic–Techniques to Sort Out the Good, the Bad, and the Ugly," www.csc.gatech.edu/~copeland/8843/slides/Analyst–011027.ppt, allegedly 2001.

Farshci, J., "Intrusion Detection FAQ, Statistical based approach to Intrusion Detection," www.sans.org/resources/idfaq/statistic ids.php, date unkoown, printed Jul. 10, 2003.

Goan, T., "A Cop on The Beat, Collecting and Appraising Intrusion Evidence," Communication of the ACM, 42(7), Jul. 1999, 46–52.

Heberlein, et al., "A Network Security Monitor," Proceedings of the IEEE Symposium on Security and Privacy, May 7–9 1990, Oakland, CA, pp. 296–304, IEEE Press.

Internet Security Systems, "Intrusion Detection for the Millennium," ISS Technology Brief, Date Unknown, pp. 1–6.

Jackson, et al., "An Expert System Application For Network Intrusion Detection," Proceedings of the 14th National Computer Security Conference, Washington, DC, Oct. 1–4, 1991.

Lankewicz, et al., "Real–time Anomaly Detection Using a Nonparametric Pattern Recognition Approach", Proceedings of the 7th Annual Computer Security Applications Conference, San Antonio, Texas, 1991, IEEE Press.

Lippmann, et al., "Evaluating Intrusion Detection Systems: The 1998 DARPA Off–line Intrusion Detection Evaluation," Proceedings of the 2000 DARPA, Information Survivability Conference and Exposition, Jan. 25–27 2000, Hilton Head, SC, vol. 2, pp. 1012–1035, IEEE Press.

Miller, L., "A Network Under Attack, Leverage Your Existing Instrumentation to Recognize and Respond to Hacker Attacks," www.netscout.com/files/Intrusion 020118.pdf, Date Unknown, pp. 1–8.

Munson, et al., "Watcher: The Missing Piece of the Security Puzzle," Proceedings of the 17th Annual Computer Security Applications Conference (ACSAC'01), Dec. 10–14 2001, New Orleans, LA, pp. 230–239, IEEE Press.

NetScreen, Products FAQ, www.netscreen.com/products/faq.html, Date Unknown.

Pearl, J., "Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference," Morgan Kaufmann Publishers, Sep. 1988.

Porras, et al., "Live Traffic Analysis of TCP/IP Gateways," Proc. 1998 ISOC Symp. On Network and Distributed Systems Security, Dec. 12, 1997, 1–13.

Skinner, "Emerald TCP Statistical Analyzer 1998 Evaluation Results," www.sdl.sri.com/emerald/98–eval–estat/index.html, Allegedly dated Jul. 9, 1999.

SRI/Stanford, "Adaptive Model–Based Monitoring and Threat Detection," Information Assurance BAA 98–34.

Staniford–Chen, et al., "GrIDS–A Graph Based Intrusion Detection System for Large Networks," Proceedings of the 19th National Information Systems Security Conference, vol. 1, pp. 361–370, Oct. 1996.

Tener, "Discovery: An Expert System in the Commercial Data Security Environment", Fourth IFIP Symposium on Information Systems Security, Monte Carlo, Dec. 1986.

Valdes, et al., "Adaptive, Model–based Monitoring for Cyber Attack Detection," Proceedings of Recent Advances in Intrusion Detection 2000 (RAID 2000), H. Debar, L. Me, F. Wu (Eds), Toulouse, France, Springer–Verlag LNCS vol. 1907, pp. 80–92, Oct. 2000.

Valdes, A., Blue Sensors, Sensor Correlation, and Alert Fusion, www.raid–symposium.org/raid2000/Materials/Abstracts/41/avaldes raidB.pdf, Oct. 4, 2000.

Valdes, et al., "Statistical Methods for Computer Usage Anomaly Detection Using NIDES (Next–Generation Intrusion Detection Expert System)," 3rd International Workshop on Rough Sets and Soft Computing, San Jose CA 1995, 306–311.

Wimer, S., "The Core of CylantSecure," White Papers, www.cylant.com/products/core.html, Date Unknown, Alleged © 1999–2003 Cylant Inc., pp. 1–4.

Zhang, et al., "A Hierarchical Anomaly Network Intrusion Detection System using Neural Network Classification," Proceedings of the 2001 WSES International Conference on Neural Networks and Applications (NNA'01), Puerto de la Cruz, Canary Islands, Spain, Feb. 11–15 2001.

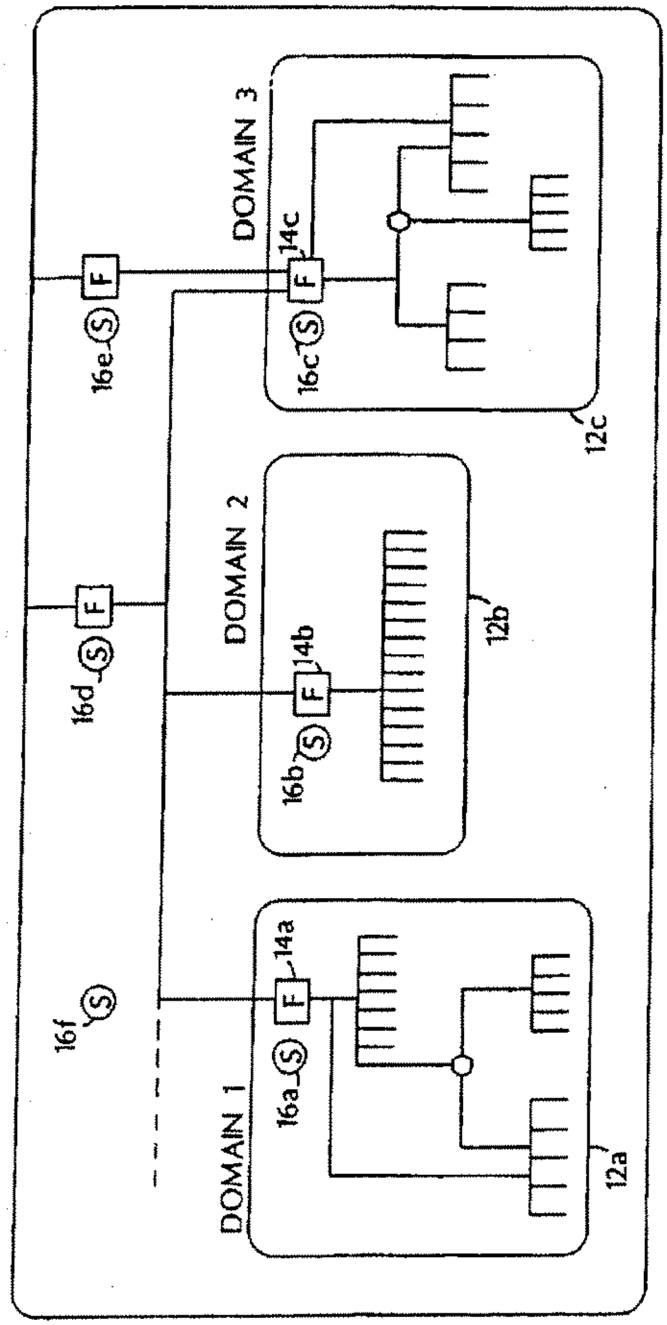

DOMAIN 3
14c
F
16c
S
12c
16e
S
F
DOMAIN 2
14b
F
16b
S
12b
16d
S
F
DOMAIN 1
14a
F
16a
S
12a
16f
S

FIG. 1

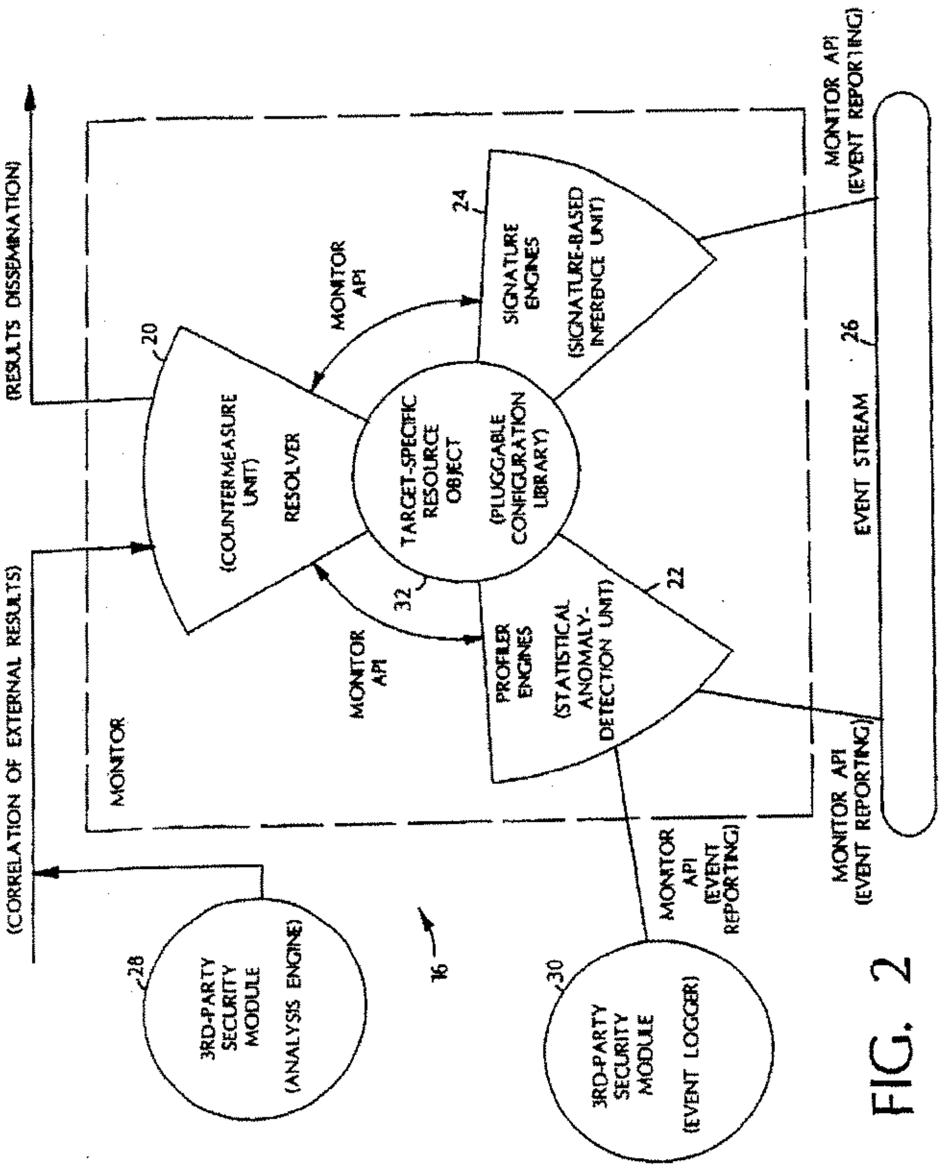

(RESULTS DISSEMINATION)
(CORRELATION OF EXTERNAL RESULTS)
MONITOR
20
(COUNTERMEASURE UNIT)
RESOLVER
TARGET-SPECIFIC RESOURCE OBJECT
(PLUGGABLE CONFIGURATION LIBRARY)
32
MONITOR API
MONITOR API
24
SIGNATURE ENGINES
(SIGNATURE-BASED INFERENCE UNIT)
22
PROFILER ENGINES
(STATISTICAL ANOMALY-DETECTION UNIT)
MONITOR API (EVENT REPORTING)
26
EVENT STREAM
MONITOR API (EVENT REPORTING)
MONITOR API (EVENT REPORTING)
28
3RD-PARTY SECURITY MODULE
(ANALYSIS ENGINE)
16
30
3RD-PARTY SECURITY MODULE
(EVENT LOGGER)

FIG. 2

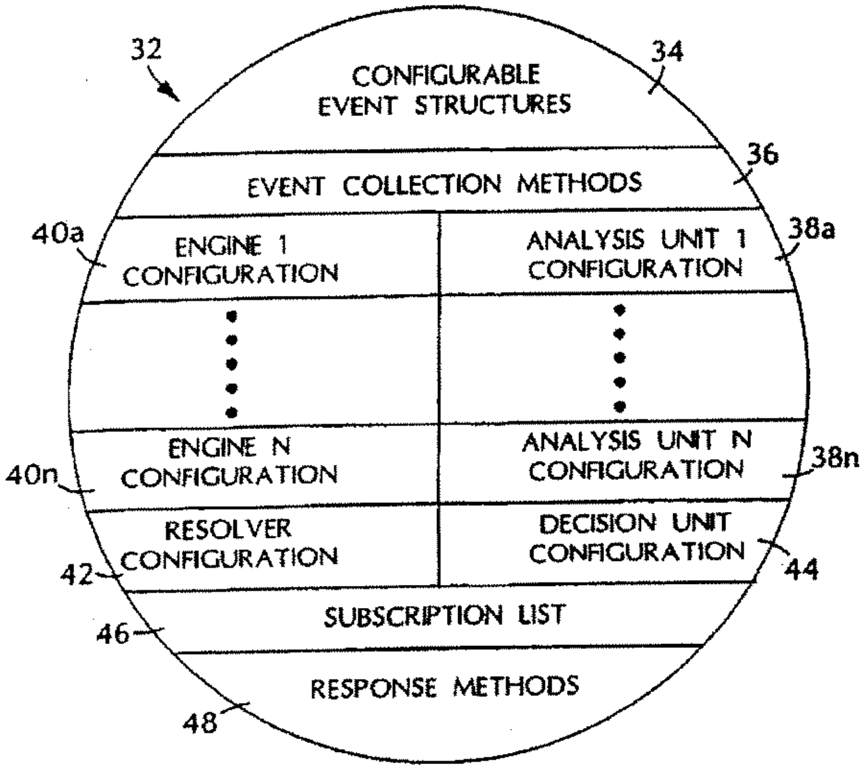

32
34
CONFIGURABLE EVENT STRUCTURES
36
EVENT COLLECTION METHODS
40a
ENGINE 1 CONFIGURATION
ANALYSIS UNIT 1 CONFIGURATION
38a
ENGINE N CONFIGURATION
ANALYSIS UNIT N CONFIGURATION
40n
38n
RESOLVER CONFIGURATION
DECISION UNIT CONFIGURATION
42
44
46
SUBSCRIPTION LIST
RESPONSE METHODS
48

FIG. 3

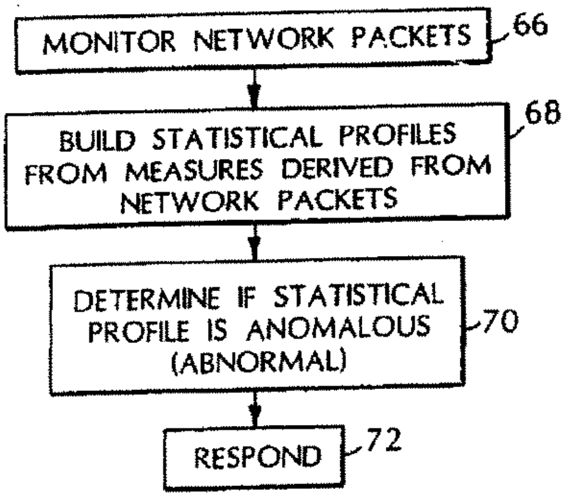

MONITOR NETWORK PACKETS
66
BUILD STATISTICAL PROFILES FROM MEASURES DERIVED FROM NETWORK PACKETS
68
DETERMINE IF STATISTICAL PROFILE IS ANOMALOUS (ABNORMAL)
70
RESPOND
72

FIG. 4

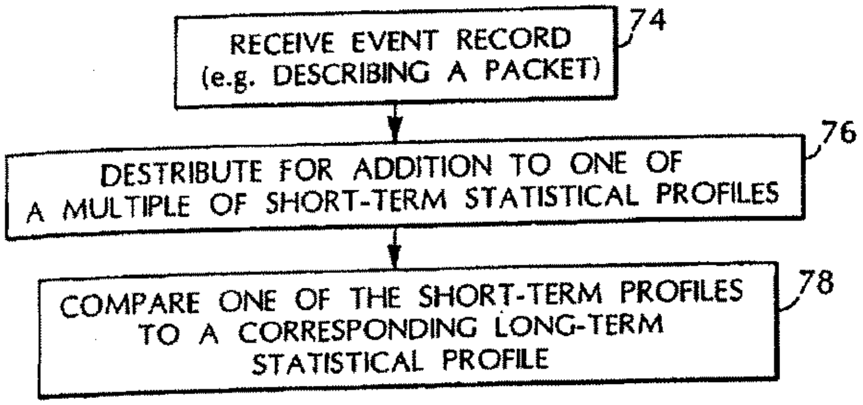

RECEIVE EVENT RECORD (e.g. DESCRIBING A PACKET)
74
DESTRIBUTE FOR ADDITION TO ONE OF A MULTIPLE OF SHORT-TERM STATISTICAL PROFILES
76
COMPARE ONE OF THE SHORT-TERM PROFILES TO A CORRESPONDING LONG-TERM STATISTICAL PROFILE
78

FIG. 5

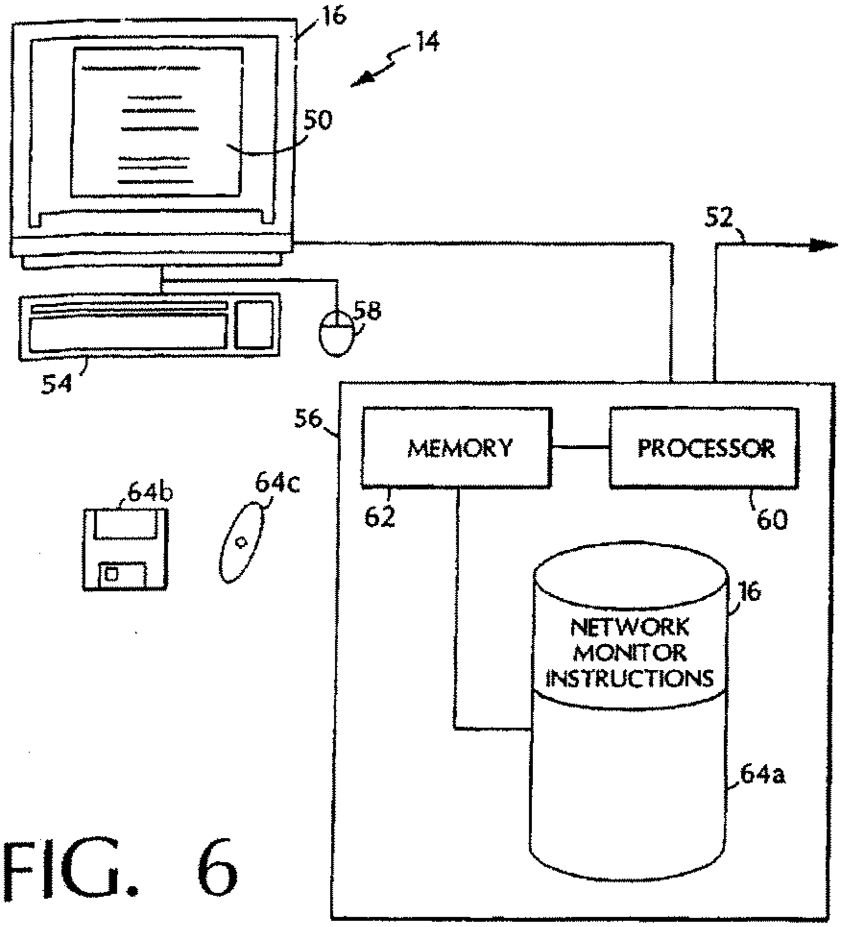

16
14
50
52
58
54
56
MEMORY
PROCESSOR
62
60
64b
64c
16
NETWORK
MONITOR
INSTRUCTIONS
64a

FIG. 6

## NETWORK SURVEILLANCE

This application is a continuation of U.S. application Ser. No. 10/254,457, filed Sep. 25, 2002, which is a continuation of U.S. application Ser. No. 09/658,137, filed Sep. 8, 2000 (now U.S. Pat. No. 6,484,203), which is a continuation of U.S. application Ser. No. 09/188,739, filed Nov. 9, 1998 (now U.S. Pat. No. 6,321,338), where all applications are herein incorporated by reference.

## REFERENCE TO GOVERNMENT FUNDING

This invention was made with Government support under Contract (Number F30602-96-C-0294 awarded by DARPA. The Government has certain rights in this invention.

## REFERENCE TO APPENDIX

An appendix consisting of 935 pages is included as part of the specification. The appendix includes material subject to copyright protection. The copyright owner does not object to the facsimile reproduction of the appendix, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights.

## BACKGROUND

The invention relates to computer networks.

Computer networks offer users ease and efficiency in exchanging information. Networks tend to include conglomerates of integrated commercial and custom-made components, interoperating and sharing information at increasing levels of demand and capacity. Such varying networks manage a growing list of needs including transportation, commerce, energy management, communications, and defense.

Unfortunately, the very interoperability and sophisticated integration of technology that make networks such valuable assets also make them vulnerable to attack, and make dependence on networks a potential liability. Numerous examples of planned network attacks, such as the Internet worm, have shown how interconnectivity can be used to spread harmful program code. Accidental outages such as the 1980 ARPAnet collapse and the 1990 AT&T collapse illustrate how seemingly localized triggering events can have globally disastrous effects on widely distributed systems. In addition, organized groups have performed malicious and coordinated attacks against various on line targets.

## SUMMARY

In general, in one aspect, a method of network surveillance includes receiving network packets (e.g., TCP/IP packets) handled by a network entity and building at least one long-term and at least one short-term statistical profile from at least one measure of the network packets that monitors data transfers, errors, or network connections. A comparison of at least one long-term and at least one short-term statistical profile is used to determine whether the difference between the short-term statistical profile and the long-term statistical profile indicates suspicious network activity.

Embodiments may include one or more of the following features. The measure may monitor data transfers by monitoring network packet data transfer commands, data transfer errors, and/or monitoring network packet data transfer volume. The measure may monitor network connections by monitoring network connection requests, network connection denials, and/or a correlation of network connections

requests and network connection denials. The measure may monitor errors by monitoring error codes included in a network packet such as privilege error codes and/or error codes indicating a reason a packet was rejected.

The method may also include responding based on the determining whether the difference between a short-term statistical profile and a long-term statistical profile indicates suspicious network activity. A response may include altering analysis of network packets and/or severing a communication channel. A response may include transmitting an event record to a network monitor, such as hierarchically higher network monitor and/or a network monitor that receives event records from multiple network monitors.

The network entity may be a gateway, a router, or a proxy server. The network entity may instead be a virtual private network entity (e.g., node).

In general, in another aspect, a method of network surveillance includes monitoring network packets handled by a network entity and building a long-term and multiple short-term statistical profiles of the network packets. A comparison of one of the multiple short-term statistical profiles with the long-term statistical profile is used to determine whether the difference between the short-term statistical profiles and the long-term statistical profile indicates suspicious network activity.

Embodiments may include one or more of the following. The multiple short-term statistical profiles may monitor different anonymous FTP sessions. Building multiple short-term statistical profiles may include deinterleaving packets to identify a short-term statistical profile.

In general, in another aspect, a computer program product, disposed on a computer readable medium, includes instructions for causing a processor to receive network packets handled by a network entity and to build at least one long-term and at least one short-term statistical profile from at least one measure of the network packets that monitors data transfers, errors, or network connections. The instructions compare a short-term and a long-term statistical profile to determine whether the difference between the short-term statistical profile and the long-term statistical profile indicates suspicious network activity.

In general, in another aspect, a method of network surveillance includes receiving packets at a virtual private network entity and statistically analyzing the received packets to determine whether the packets indicate suspicious network activity. The packets may or may not be decrypted before statistical analysis

Advantages may include one or more of the following. Using long-term and a short-term statistical profiles from measures that monitor data transfers, errors, or network connections protects network components from intrusion. As long-term profiles represent "normal" activity, abnormal activity may be detected without requiring an administrator to catalog each possible attack upon a network. Additionally, the ability to deinterleave packets to create multiple short-term profiles for comparison against a long-term profile enables the system to detect abnormal behavior that may be statistically ameliorated if only a single short-term profile was created.

The scheme of communication network monitors also protects networks from more global attacks. For example, an attack made upon one network entity may cause other entities to be alerted. Further, a monitor that collects event reports from different monitors may correlate activity to identify attacks causing disturbances in more than one network entity.

Additionally, statistical analysis of packets handled by a virtual private network enable detection of suspicious network activity despite virtual private network security techniques such as encryption of the network packets.

Other features and advantages will become apparent from the following description, including the drawings, and from the claims.

## BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a diagram of network monitors deployed in an enterprise.

FIG. 2 is a diagram of a network monitor that monitors an event stream.

FIG. 3 is a diagram of a resource object that configures the network monitor of FIG. 2

FIG. 4 is a flowchart illustrating network surveillance.

FIG. 5 is a flowchart illustrating multiple short-term statistical profiles for comparison against a single long-term statistical profile.

FIG. 6 diagram of a computer platform suitable for deployment of a network monitor.

## DETAILED DESCRIPTION

Referring to FIG. 1, an enterprise 10 includes different domains 12a–12c. Each domain 12a–12c includes one or more computers offering local and network services that provide an interface for requests internal and external to the domain 12a–12c. Network services include features common to many network operating systems such as mail, HTTP, FTP, remote login, network file systems, finger, Kerberos, and SNMP. Some domains 12a–12c may share trust relationships with other domains (either peer-to-peer or hierarchical). Alternatively, domains 12a–12c may operate in complete mistrust of all others, providing outgoing connections only or severely restricting incoming connections. Users may be local to a single domain or may possess accounts on multiple domains that allow them to freely establish connections throughout the enterprise 10.

As shown, the enterprise 10 includes dynamically deployed network monitors 16a–16f that analyze and respond to network activity and can interoperate to form an analysis hierarchy. The analysis hierarchy provides a framework for the recognition of more global threats to interdomain connectivity, including coordinated attempts to infiltrate or destroy connectivity across an entire network enterprise 10. The hierarchy includes service monitors 16a–16c, domain monitors 16d–16e, and enterprise monitors 16f.

Service monitors 16a–16c provide local real-time analysis of network packets (e.g., TCP/IP packets) handled by a network entity 14a–14c. Network entities include gateways, routers, firewalls, or proxy servers. A network entity may also be part of a virtual private network. A virtual private network (VPN) is constructed by using public wires to connect nodes. For example, a network could use the Internet as the medium for transporting data and use encryption and other security mechanisms to ensure that only authorized users access the network and that the data cannot be intercepted. A monitor 16a–16f can analyze packets both before and after decryption by a node of the virtual private network.

Information gathered by a service monitor 16a–16c can be disseminated to other monitors 16a–16f, for example, via a subscription-based communication scheme. In a subscription-based scheme client monitors subscribe to receive analysis reports produced by server monitors. As a monitor 16a–16f produces analysis reports, the monitor 16a–16f disseminates these reports asynchronously to subscribers. Through subscription, monitors 16d–16f distributed throughout a large network are able to efficiently disseminate reports of malicious activity without requiring the overhead of synchronous polling.

Domain monitors 16d–16e perform surveillance over all or part of a domain 12a–12c. Domain monitors 16d–16e correlate intrusion reports disseminated by individual service monitors 16a–16c, providing a domain-wide perspective of activity (or patterns of activity). In addition to domain surveillance, domain monitors 16a–16c can reconfigure system parameters, interface with other monitors beyond a domain, and report threats against a domain 12a–12c to administrators. Domain monitors 16d–16e can subscribe to service monitors 16a–16c. Where mutual trust among domains 12a–12c exists, domain monitors 16d–16e may establish peer relationships with one another. Peer-to-peer subscription allows domain monitors 16d–16e to share analysis reports produced in other domains 12a–12c. Domain monitors 16d–16e may use such reports to dynamically sensitize their local service monitors 16a–16c to malicious activity found to be occurring outside a domain 12a–12c. Domain monitors 16d–16e may also operate within an enterprise hierarchy where they disseminate analysis reports to enterprise monitors 16f for global correlation.

Enterprise monitors 16f correlate activity reports produced across the set of monitored domains 12a–12c. Enterprise 10 surveillance may be used where domains 12a–12c are interconnected under the control of a single organization, such as a large privately owned WAN (Wide Area Network). The enterprise 10, however, need not be stable in its configuration or centrally administered. For example, the enterprise 10 may exist as an emergent entity through new interconnections of domains 12a–12c. Enterprise 10 surveillance is very similar to domain 12a–12c surveillance: an enterprise monitor 16f subscribes to various domain monitors 16d–16e, just as the domain monitors 16d–16e subscribed to various service monitors 16a–16c. The enterprise monitor 16f (or monitors, as it would be important to avoid centralizing any analysis) focuses on network-wide threats such as Internet worm-like attacks, attacks repeated against common network services across domains, or coordinated attacks from multiple domains against a single domain. As an enterprise monitor 16f recognizes commonalities in intrusion reports across domains (e.g., the spreading of a worm or a mail system attack repeated throughout the enterprise 10), the monitor 16f can help domains 12a–12c counter the attack and can sensitize other domains 12a–12c to such attacks before they are affected. Through correlation and sharing of analysis reports, reports of problems found by one monitor 16a–16f may propagate to other monitors 16a–16f throughout the network. Interdomain event analysis is vital to addressing more global, information attacks against the entire enterprise 10.

Referring to FIG. 2, each monitor 16 includes one or more analysis engines 22, 24. These engines 22, 24 can be dynamically added, deleted, and modified as necessary. In the dual-analysis configuration shown, a monitor 16 instantiation includes a signature analysis engine 22 and a statistical profiling engine 24. In general, a monitor 16 may include additional analysis engines that may implement other forms of analysis. A monitor 16 also includes a resolver 20 that implements a response policy and a resource object 32 that configures the monitor 16. The monitors 16 incorporate an application programmers' interface (API) that enhances encapsulation of monitor functions and eases integration of thir-party intrusion-detection tools **28, 30**.

Each monitor **16** can analyze event records that form an event stream. The event stream may be derived from a variety of sources such as TCP/IP network packet contents or event records containing analysis reports disseminated by other monitors. For example, an event record can be formed from data included in the header and data segment of a network packet. The volume of packets transmitted and received, however, dictates careful assessment of ways to select and organize network packet information into event record streams.

Selection of packets can be based on different criteria. Streams of event records can be derived from discarded traffic (i.e., packets not allowed through the gateway because they violate filtering rules), pass-through traffic (i.e., packets allowed into the internal network from external sources), packets having a common protocol (e.g., all ICMP (Internet. Control Message Protocol) packets that reach the gateway), packets involving network connection management (e.g., SIN, RESET, ACK, [window resize]), and packets targeting ports to which an administrator has not assigned any network service and that also remain unblocked by the firewall. Event streams may also be based on packet source addresses (e.g., packets whose source addresses match well-known external sites such as satellite offices or have raised suspicion from other monitoring efforts) or destination addresses (e.g., packets whose destination addresses match a given internal host or workstation). Selection can also implement application-layer monitoring (e.g., packets targeting a particular network service or application). Event records can also be produced from other sources of network packet information such as report logs produced by network entities. Event streams can be of very fine granularity. For example, a different stream might be derived for commands received from different commercial web-browsers since each web-browser produces different characteristic network activity.

A monitor **16** can also construct interval summary event records, which contain accumulated network traffic statistics (e.g., number of packets and number of kilobytes transferred). These event records are constructed at the end of each interval (e.g., once per N seconds). Event records are forwarded to the analysis engines **22, 24** for analysis.

The profile engine **22** can use a wide range of multivariate statistical measures to profile network activity indicated by an event stream. A statistical score represents how closely currently observed usage corresponds to the established patterns of usage. The profiler engine **22** separates profile management and the mathematical algorithms used to assess the anomaly of events. The profile engine **22** may use a statistical analysis technique described in A. Valdes and D. Anderson, "Statistical Methods for Computer Usage Anomaly Detection Using NIDES", Proceedings of the Third International Workshop on Rough Sets and Soft Computing, January 1995, which is incorporated by reference in its entirety. Such an engine **22** can profile network activity via one or more variables called measures. Measures can be categorized into four classes: categorical, continuous, intensity, and event distribution measures.

Categorical measures assume values from a discrete, nonordered set of possibilities. Examples of categorical measures include network source and destination addresses, commands (e.g., commands that control data transfer and manage network connections), protocols, error codes (e.g., privilege violations, malformed service requests, and mal-

formed packet codes), and port identifiers. The profiler engine **22** can build empirical distributions of the category values encountered, even if the list of possible values is open-ended. The engine **22** can have mechanisms for "aging out" categories whose long-term probabilities drop below a threshold.

Continuous measures assume values from a continuous or ordinal set. Examples include inter-event time (e.g., difference in time stamps between consecutive events from the same stream), counting measures such as the number of errors of a particular type observed in the recent past, the volume of data transfers over a period of time, and network traffic measures (number of packets and number of kilobytes). The profiler engine **22** treats continuous measures by first allocating bins appropriate to the range of values of the underlying measure, and then tracking the frequency of observation of each value range. In this way, multi-modal distributions are accommodated and much of the computational machinery used for categorical measures is shared. Continuous measures are useful not only for intrusion detection, but also to support the monitoring of the health and status of the network from the perspective of connectivity and throughput. For example, a measure of traffic volume maintained can detect an abnormal loss in the data rate of received packets when this volume falls outside historical norms. This sudden drop can be specific both to the network entity being monitored and to the time of day (e.g., the average sustained traffic rate for a major network artery is much different at 11:00 a.m. than at midnight).

Intensity measures reflect the intensity of the event stream (e.g., number of ICMP packets) over specified time intervals (e.g., 1 minute, 10 minutes, and 1 hour). Intensity measures are particularly suited for detecting flooding attacks, while also providing insight into other anomalies.

Event distribution measures are met-measures that describes how other measures in the profile are affected by each event. For example, an "ls" command in an FTP session affects the directory measure, but does not affect measures related to file transfer. This measure is not interesting for all event streams. For example, all network-traffic event records affect the same measures (number of packets and kilobytes) defined for that event stream, so the event distribution does not change. On the other hand, event distribution measures are useful in correlative analysis performed by a monitor **16*a*–16*f*** that receives reports from other monitors **16*a*–16*f***.

The system maintains and updates a description of behavior with respect to these measure types in an updated profile. The profile is subdivided into short-term and long-term profiles. The short-term profile accumulates values between updates, and exponentially ages (e.g., weighs data based on how long ago the data was collected) values for comparison to the long-term profile. As a consequence of the aging mechanism, the short-term profile characterizes recent activity, where "recent" is determined by a dynamically configurable aging parameters. At update time (typically, a time of low system activity), the update function folds the short-term values observed since the last update into the long-term profile, and the short-term profile is cleared. The long-term profile is itself slowly aged to adapt to changes in subject activity. Anomaly scoring compares related attributes in the short-term profile against the long-term profile. As all evaluations are done against empirical distributions, no assumptions of parametric distributions are made, and multi-modal and categorical distributions are accommodated. Furthermore, the algorithms require no a priori knowledge of intrusive or exceptional activity.

The statistical algorithm adjusts a short-term profile for the measure values observed in the event record. The distribution of recently observed values is compared against the long-term profile, and a distance between the two is obtained. The difference is compared to a historically adaptive deviation. The empirical distribution of this deviation is transformed to obtain a score for the event. Anomalous events are those whose scores exceed a historically adaptive score threshold based on the empirical score distribution. This nonparametric approach handles all measure types and makes no assumptions on the modality of the distribution for continuous measures.

Profiles are provided to the computational engine as classes defined in the resource object 32. The mathematical functions for anomaly scoring, profile maintenance, and updating do not require knowledge of the data being analyzed beyond what is encoded in the profile class. Event collection interoperability supports translation of the event stream to the profile and measure classes. At that point, analysis for different types of monitored entities is mathematically similar. This approach imparts great flexibility to the analysis in that fading memory constants, update frequency, measure type, and so on are tailored to the network entity being monitored.

The measure types described above can be used individually or in combination to detect network packet attributes characteristic of intrusion. Such characteristics include large data transfers (e.g., moving or downloading files), an increase in errors (e.g., an increase in privilege violations or network packet rejections), network connection activity, and abnormal changes in network volume.

As shown, the monitor 16 also includes a signature engine 24. The signature engine 24 maps an event stream against abstract representations of event sequences that are known to indicate undesirable activity. Signatu-reanalysis objectives depend on which layer in the hierarchical analysis scheme the signature engine operates. Service monitor 16*a*–16*c* signature engines 24 attempt to monitor for attempts to penetrate or interfere with the domain's operation. The signature engine scans the event stream for events that represent attempted exploitations of known attacks against the service, or other activity that stands alone as warranting a response from the monitor. Above the service layer, signature engines 24 scan the aggregate of intrusion reports from service monitors in an attempt to detect more global coordinated attack scenarios or scenarios that exploit interdependencies among network services. Layering signature engine analysis enables the engines 24 to avoid misguided searches along incorrect signature paths in addition to distributing the signature analysis.

A signature engines 24 can detect, for example, address spoofing, tunneling, source routing, SATAN attacks, and abuse of ICMP messages ("Redirect" and "Destination Unreachable" messages in particular). Threshold analysis is a rudimentary, inexpensive signature analysis technique that records the occurrence of specific events and, as the name implies, detects when the number of occurrences of that event surpasses a reasonable count. For example, monitors can encode thresholds to monitor activity such as the number of fingers, pings, or failed login requests to accounts such as guest, demo, visitor, anonymous FTP, or employees who have departed the company.

Signature engine 24 can also examine the data portion of packets in search of a variety of transactions that indicate suspicious, if not malicious, intentions by an external client. The signature engine 24, for example, can parse FTP traffic traveling through the firewall or router for unwanted transfers of configuration or specific system data, or anonymous requests to access non-public portions of the directory structure. Similarly, a monitor can analyze anonymous FTP sessions to ensure that the file retrievals and uploads/modifications are limited to specific directories. Additionally, signature analysis capability can extend to session analyses of complex and dangerous, but highly useful, services like HTTP or Gopher.

Signature analysis can also scan traffic directed at unused ports (i.e., ports to which the administrator has not assigned a network service). Here, packet parsing can be used to study network traffic after some threshold volume of traffic, directed at an unused port, has been exceeded. A signature engine 24 can also employ a knowledge base of known telltale packets that are indicative of well-known network-service protocol traffic (e.g., FTP, Telnet, SMTP, HTTP). The signature engine 24 then determines whether the unknown port traffic matches any known packet sets. Such comparisons could lead to the discovery of network services that have been installed without an administrator's knowledge.

The analysis engines 22, 24 receive large volumes of events and produce smaller volumes of intrusion or suspicion reports that are then fed to the resolver 20. The resolver 20 is an expert system that receives the intrusion and suspicion reports produced by the analysis engines 22, 24 and reports produced externally by other analysis engines to which it subscribes. Based on these reports, the resolver 20 invokes responses. Because the volume of intrusion and suspicion reports is lower than the volume of events received by the analysis engines 22, 24, the resolver 20 can afford the more sophisticated demands of configuration maintenance and managing the response handling and external interfaces necessary for monitor operation. Furthermore, the resolver 20 adds to extensibility by providing the subscription interface through which third-party analysis tools 28, 30 can interact and participate in the hierarchical analysis scheme.

Upon its initialization, the resolver 20 initiates authentication and subscription sessions with those monitors 16*a*–16*f* whose identities appear in the monitor's 16 subscription-list (46 FIG. 3). The resolver 20 also handles all incoming requests by subscribers, which must authenticate themselves to the resolver 20. Once a subscription session is established with a subscriber monitor, the resolver 20 acts as the primary interface through which configuration requests are received and intrusion reports are disseminated.

Thus, resolvers 20 can request and receive reports from other resolvers at lower layers in the analysis hierarchy. The resolver 20 forwards analysis reports received from subscribers to the analysis engines 22, 24. This tiered collection and correlation of analysis results allows monitors 16*a*–16*f* to represent and profile global malicious or anomalous activity that is not visible locally.

In addition to external-interface responsibilities, the resolver 20 operates as a fully functional decision engine, capable of invoking real-time response measures in response to malicious or anomalous activity reports produced by the analysis engines. The resolver 20 also operates as the center of intramonitor communication. As the analysis engines 22, 24 build intrusion and suspicion reports, they propagate these reports to the resolver 20 for further correlation, response, and dissemination to other monitors 16*a*–16*f*. The resolver 20 can also submit runtime configuration requests to the analysis engines 22, 24, for example, to increase or decrease the scope of analyses (e.g., enable or disable additional signature rules) based on various operating metrics. These configuration requests could be made as a result of encountering other intrusion reports from other subscribers. For example, a report produced by a service monitor 16a–16c in one domain could be propagated to an enterprise monitor 16f, which in turn sensitizes service monitors in other domains to the same activity.

The resolver 20 also operates as the interface mechanism between administrators and the monitor 16. From the perspective of a resolver 20, the administrator interface is simply a subscribing service to which the resolver 20 may submit reports and receive configuration requests. An administrative interface tool can dynamically subscribe and unsubscribe to any of the deployed resolvers 20, as well as submit configuration requests and asynchronous probes as desired.

The monitors 16a–16f incorporate a bidirectional messaging system that uses a standard interface specification for communication within and between monitor elements and external modules. Using this interface specification, third-party modules 28, 30 can communicate with monitors. For example, third-party modules 28 can submit event records to the analysis engines 22, 24 for processing. Additionally, third-party modules 30 may also submit and receive analysis results via the resolver's 20 external interfaces. Thus, third-party modules 28, 30 can incorporate the results from monitors into other surveillance efforts or contribute their results to other monitors 16a–16f. Lastly, the monitor's 16 internal API allows third-party analysis engines to be linked directly into the monitor boundary.

The message system operates under an asynchronous communication model for handling results dissemination and processing that is generically referred to as subscription-based message passing. Component interoperation is client/server-based, where a client module may subscribe to receive event data or analysis results from servers. Once a subscription request is accepted by the server, the server module forwards events or analysis results to the client automatically as data becomes available, and may dynamically reconfigure itself as requested by the client's control requests. This asynchronous model reduces the need for client probes and acknowledgments.

The interface supports an implementation-neutral communication framework that separates the programmer's interface specification and the issues of message transport. The interface specification embodies no assumptions about implementation languages, host platform, or a network. The transport layer is architecturally isolated from the internals of the monitors so that transport modules may be readily introduced and replaced as protocols and security requirements are negotiated between module developers. The interface specification involves the definition of the messages that the various intrusion-detection modules must convey to one another and how these messages should be processed. The message structure and content are specified in a completely implementation-neutral context.

Both intramonitor and intermonitor communication employ identical subscription-based client-server models. With respect to intermonitor communication, the resolver 20 operates as a client to the analysis engines, and the analysis engines 22, 24 operate as clients to the event filters. Through the internal message system, the resolver 20 submits configuration requests to the analysis engines 22, 24, and receives from the analysis engines 22, 24 their analysis results. The analysis engines 22, 24 operate as servers providing the resolver 20 with intrusion or suspicion reports either asynchronously or upon request. Similarly, the analysis engines 22, 24 are responsible for establishing and maintaining a communication link with an event collection method (or event filter) and prompting the reconfiguration of the collection method's filtering semantics when necessary.

Intermonitor communication also operates using the subscription-based hierarchy. A domain monitor 16d–16e subscribes to the analysis results produced by service monitors 16a–16c, and then propagates its own, analytical reports to its parent enterprise monitor 16f. The enterprise monitor 16f operates as a client to one or more domain monitors 16d–16e, allowing them to correlate and model enterprise-wide activity from the domain-layer results. Domain monitors 16d–16e operate as servers to the enterprise monitors 16f, and as clients to the service monitors 16a–16c deployed throughout their domain 12a–12c. This message scheme can operate substantially the same if correlation were to continue at higher layers of abstraction beyond enterprise 10 analysis.

Intramonitor and intermonitor programming interfaces are substantially the same. These interfaces can be subdivided into five categories of interoperation: channel initialization and termination, channel synchronization, dynamic configuration, server probing, and report/event dissemination. Clients are responsible for initiating and terminating channel sessions with servers. Clients are also responsible for managing channel synchronization in the event of errors in message sequencing or periods of failed or slow response (i.e., "I'm alive" confirmations). Clients may also submit dynamic configuration requests to servers. For example, an analysis engine 22, 24 may request an event collection method to modify its filtering semantics. Clients may also probe servers for report summaries or additional event information. Lastly, servers may send clients intrusion/suspicion reports in response to client probes or in an asynchronous dissemination mode.

The second part of the message system framework involves specification of a transport mechanism used to establish a given communication channel between monitors 16a–16f or possibly between a monitor 16a–16f and a third-party security module. All implementation dependencies within the message system framework are addressed by pluggable transport modules. Transport modules are specific to the participating intrusion-detection modules, their respective hosts, and potentially to the network—should the modules require cross-plafform interoperation. Instantiating a monitor 16a–16f may involve incorporation of the necessary transport module(s) (for both internal and external communication) The transport modules that handle intramonitor communication may be different from the transport modules that handle intermonitor communication. This allows the intramonitor transport modules to address security and reliability issues differently than how the intermonitor transport modules address security and reliability. While intramonitor communication may more commonly involve interprocess communication within a single host, intermonitor communication will most commonly involve cross-platform networked interoperation. For example, the intramonitor transport mechanisms may employ unnamed pipes which provides a kernel-enforced private interprocess communication channel between the monitor 16 components (this assumes a process hierarchy within the monitor 16 architecture). The monitor's 16 external transport, however, will more likely export data through untrusted network connections and thus require more extensive security management. To ensure the security and integrity of the message exchange, the external transport may employ public/private key authentication protocols and session key exchange. Using this same interface, third-party analysis tools may authenticate and exchange analysis results and configuration information in a well-defined, secure manner.

The pluggable transport permits flexibility in negotiating security features and protocol usage with third parties. Incorporation of a commercially available network management system can deliver monitoring results relating to security, reliability, availability, performance, and other attributes. The network management system may in turn subscribe to monitor produced results in order to influence network reconfiguration.

All monitors (service, domain, and enterprise) 16*a*–16*f* use the same monitor code-base. However, monitors may include different resource objects 32 having different configuration data and methods. This reusable software architecture can reduce implementation and maintenance efforts. Customizing and dynamically configuring a monitor 16 thus becomes a question of building and/or modifying the resource object 32.

Referring to FIG. 3, the resource object 32 contains the operating parameters for each of the monitor's 16 components as well as the analysis semantics (e.g., the profiler engine's 22 measure and category definition, or the signature engine's 24 penetration rule-base) necessary to process an event stream. After defining a resource object 32 to implement a particular set of analyses on an event stream, the resource object 32 may be reused by other monitors 16 deployed to analyze equivalent event streams. For example, the resource object 32 for a domain's router may be reused as other monitors 16 are deployed for other routers in a domain 12*a*–12*c*. A library of resource objects 32 provides prefabricated resource objects 32 for commonly available network entities.

The resource object 32 provides a pluggable configuration module for tuning the generic monitor code-base to a specific event stream. The resource object 32 includes configurable event structures 34, analysis unit configuration 38*a*–38*n*, engine configuration 40*a*–40*n*, resolver configuration 42, decision unit configuration 44, subscription list data 46, and response methods 48.

Configurable event structures 34 define the structure of event records and analysis result records. The monitor code-base maintains no internal dependence on the content or format of any given event stream or the analysis results produced from analyzing the event stream. Rather, the resource object 32 provides a universally applicable syntax for specifying the structure of event records and analysis results. Event records are defined based on the contents of an event stream(s). Analysis result structures are used to package the findings produced by analysis engines. Event records and analysis results are defined similarly to allow the eventual hierarchical processing of analysis results as event records by subscriber monitors.

Event-collection methods 36 gather and parse event records for analysis engine processing. Processing by analysis engines is controlled by engine configuration 40*a*–40*n* variables and data structures that specify the operating configuration of a fielded monitor's analysis engine(s). The resource object 32 maintains a separate collection of operating parameters for each analysis engine instantiated in the monitor 16. Analysis unit configuration 38*a*–38*n* include configuration variables that define the semantics employed by the analysis engine to process the event stream.

The resolver configuration 42 includes operating parameters that specify the configuration of the resolver's internal modules. The decision unit configuration 44 describes semantics used by the resolver's decision unit for merging the analysis results from the various analysis engines. The semantics include the response criteria used to invoke countermeasure handlers. A resource object 32 may also include response methods 48. Response methods 48 include preprogrammed countermeasure methods that the resolver may invoke as event records are received. A response method 48 includes evaluation metrics for determining the circumstances under which the method should be invoked. These metrics include a threshold metric that corresponds to the measure values and scores produced by the profiler engine 22 and severity metrics that correspond to subsets of the associated attack sequences defined within the resource object 32.

Countermeasures range from very passive responses, such as report dissemination to other monitors 16*a*–16*f* or administrators, to highly aggressive actions, such as severing a communication channel or the reconfiguration of logging facilities within network components (e.g., routers, firewalls, network services, audit daemons). An active response may invoke handlers that validate the integrity of network services or other assets to ensure that privileged network services have not been subverted. Monitors 16*a*–16*f* may invoke probes in an attempt to gather as much counterintelligence about the source of suspicious traffic by using features such as traceroute or finger.

The resource object 32 may include a subscription list 46 that includes information necessary for establishing subscription-based communication sessions, which may include network address information and public keys used by the monitor to authenticate potential clients and servers. The subscription list 46 enables transmission or reception of messages that report malicious or anomalous activity between monitors. The most obvious examples where relationships are important involve interdependencies among network services that make local policy decisions. For example, the interdependencies between access checks performed during network file system mounting and the IP mapping of the DNS service. An unexpected mount monitored by the network file system service may be responded to differently if the DNS monitor informs the network file system monitor of suspicious updates to the mount requestor's DNS mapping.

The contents of the resource object 32 are defined and utilized during monitor 16 initialization. In addition, these fields may be modified by internal monitor 16 components, and by authorized external clients using the monitor's 16 API. Modifying the resource object 32 permits adaptive analysis of an event stream, however, it also introduces a potential stability problem if dynamic modifications are not tightly restricted to avoid cyclic modifications. To address this issue, monitors 16 can be configured to accept configuration requests from only higher-level monitors 16.

Referring to FIG. 4, a monitor performs network surveillance by monitoring 66 a stream of network packets. The monitor builds a statistical model of network activity from the network packets, for example, by building 68 long-term and short-term statistical profiles from measures derived from the network packets. The measures include measures that can show anomalous network activity characteristic of network intrusion such as measures that describe data transfers, network connections, privilege and network errors, and abnormal levels of network traffic. The monitor can compare 70 the long-term and short-term profiles to detect suspicious network activity. Based on this comparison, the monitor can respond 72 by reporting the activity to another monitor or by executing a countermeasure response. More information can be found in P. Porras and A. Valdes "Live Traffic Analysis of TCP/IP Gateways", Networks and Distributed Systems Security Symposium, March 1998, which is incorporated by reference in its entirety.

A few examples can illustrate this method of network surveillance. Network intrusion frequently causes large data transfers, for example, when an intruder seeks to download sensitive files or replace system files with harmful substitutes. A statistical profile to detect anomalous data transfers might include a continuous measure of file transfer size, a categorical measure of the source or destination directory of the data transfer, and an intensity measure of commands corresponding to data transfers (e.g., commands that download data). These measures can detect a wide variety of data transfer techniques such as a large volume of small data transfers via e-mail or downloading large files en masse. The monitor may distinguish between network packets based on the time such packets were received by the network entity, permitting statistical analysis to distinguish between a normal data transfer during a workday and an abnormal data transfer on a weekend evening.

Attempted network intrusion may also produce anomalous levels of errors. For example, categorical and intensity measures derived from privilege errors may indicate attempts to access protected files, directories, or other network assets. Of course, privilege errors occur during normal network operation as users mistype commands or attempt to perform an operation unknowingly prohibited. By comparing the long-term and short-term statistical profiles, a monitor can distinguish between normal error levels and levels indicative of intrusion without burdening a network administrator with the task of arbitrarily setting an unvarying threshold. Other measures based on errors, such as codes describing why a network entity rejected a network packet enable a monitor to detect attempts to infiltrate a network with suspicious packets.

Attempted network intrusion can also be detected by measures derived from network connection information. For example, a measure may be formed from the correlation (e.g., a ratio or a difference) of the number of SYN connection request messages with the number of SIN_ACK connection acknowledgment messages and/or the number of ICMP messages sent. Generally, SIN requests received should balance with respect to the total of SIN_ACK and ICMP messages sent. That is, flow into and out-of a network entity should be conserved. An imbalance can indicate repeated unsuccessful attempts to connect with a system, perhaps corresponding to a methodical search for an entry point to a system. Alternatively, intensity measures of transport-layer connection requests, such as a volume analysis of SYN-RST messages, could indicate the occurrence of a SIN-attack against port availability or possibly port-scanning. Variants of this can include intensity measures of TCP/FIN messages, considered a more stealthy form of port scanning.

Many other measures can detect network intrusion. For example, "doorknob rattling," testing a variety of potentially valid commands to gain access (e.g., trying to access a "system" account with a password of "system"), can be detected by a variety of categorical measures. A categorical measure of commands included in network packets can identify an unusual short-term set of commands indicative of "doorknob-rattling." Similarly, a categorical measure of protocol requests may also detect an unlikely mix of such requests.

Measures of network packet volume can also help detect malicious traffic, such as traffic intended to cause service denials or perform intelligence gathering, where such traffic may not necessarily be violating filtering policies. A measure reflecting a sharp increase in the overall volume of discarded packets as well as a measure analyzing the disposition of the discarded packets can provide insight into unintentionally malformed packets resulting from poor line quality or internal errors in neighboring hosts. High volumes of discarded packets can also indicate more maliciously intended transmissions such as scanning of UPD ports or IP address scanning via ICMP echoes. Excessive number of mail expansion request commands.(EXPN) may indicate intelligence gathering, for example, by spammers.

A long-term and short-term statistical profile can be generated for each event stream. Thus, different event streams can "slice" network packet data in different ways. For example, an event stream may select only network packets having a source address corresponding to a satellite office. Thus, a long-term and short-term profile will be generated for the particular satellite office. Thus, although a satellite office may have more privileges and should be expected to use more system resources than other external addresses, a profile of satellite office use can detect "address spoofing" (i.e., modifying packet information to have a source address of the satellite office).

The same network packet event may produce records in more than one event stream. For example, one event stream may monitor packets for FTP commands while another event stream monitors packets from a particular address. In this case, an FTP command from the address would produce an event record in each stream.

Referring to FIG. 5, a monitor may also "deinterleave." That is, the monitor may create and update 74, 76 more than one short-term profile for comparison 78 against a single long-term profile by identifying one of the multiple short-term profiles that will be updated by an event record in an event stream. For example, at any one time a network entity may handle several FTP "anonymous" sessions. If each network packet for all anonymous sessions were placed in a single short-term statistical profile, potentially intrusive activity of one anonymous session may be statistically ameliorated by non-intrusive sessions. By creating and updating short-term statistical profiles for each anonymous session, each anonymous session can be compared against the long-term profile of a normal FTP anonymous session. Deinterleaving can be done for a variety of sessions including HTTP sessions (e.g., a short-term profile for each browser session).

Referring to FIG. 6, a computer platform 14 suitable for executing a network monitor 16 includes a display 50, a keyboard 54, a pointing device 58 such as a mouse, and a digital computer 56. The digital computer 56 includes memory 62, a processor 60, a mass storage device 64*a*, and other customary components such as a memory bus and peripheral bus. The platform 14 may further include a network connection 52.

Mass storage device 64*a* can store instructions that form a monitor 16. The instructions may be transferred to memory 62 and processor 60 in the course of operation. The instructions 16 can cause the display 50 to display images via an interface such as a graphical user interface. Of course, instructions may be stored on a variety of mass storage devices such as a floppy disk 64*b*, CD-ROM 640, or PROM (not shown).

Other embodiments are within the scope of the following claims.

What is claimed is:

1. Method for monitoring an enterprise network, said method comprising the steps of:

deploying a plurality of network monitors in the enterprise network;

detecting, by the network monitors, suspicious network activity based on analysis of network traffic data, wherein at least one of the network monitors utilizes a statistical detection method;

generating, by the monitors, reports of said suspicious activity; and

automatically receiving and integrating the reports of suspicious activity, by one or more hierarchical monitors.

2. The method of claim 1, wherein at least one of the network monitors utilizes a signature matching detection method.

3. The method of claim 2, wherein the monitor utilizing a signature matching detection method also utilizes a statistical detection method.

4. The method of claim 1, wherein integrating comprises correlating intrusion reports reflecting underlying commonalities.

5. The method of claim 1, wherein integrating further comprises invoking countermeasures to a suspected attack.

6. The method of claim 1, wherein the plurality of network monitors includes an API for encapsulation of monitor functions and integration of third-party tools.

7. The method of claim 1, wherein the enterprise network is a TCP/IP network.

8. The method of claim 1, wherein the network monitors are deployed at one or more of the following facilities of the enterprise network: {gateways, routers, proxy servers}.

9. The method of claim 1, wherein deploying the network monitors includes placing a plurality of service monitors among multiple domains of the enterprise network.

10. The method of claim 9, wherein receiving and integrating is performed by a domain monitor with respect to a plurality of service monitors within the domain monitor's associated network domain.

11. The method of claim 1, wherein deploying the network monitors includes deploying a plurality of domain monitors within the enterprise network, each domain monitor being associated with a corresponding domain of the enterprise network.

12. The method of claim 11, wherein receiving and integrating is performed by an enterprise monitor with respect to a plurality of domain monitors within the enterprise network.

13. The method of claim 11, wherein the plurality of domain monitors within the enterprise network establish peer-to-peer relationships with one another.

14. An enterprise network monitoring system comprising:

a plurality of network monitors deployed within an enterprise network, said plurality of network monitors detecting suspicious network activity based on analysis of network traffic data, wherein at least one of the network monitors utilizes a statistical detection method;

said network monitors generating reports of said suspicious activity; and

one or more hierarchical monitors in the enterprise network, the hierarchical monitors adapted to automatically receive and integrate the reports of suspicious activity.

15. The system of claim 14, wherein the integration comprises correlating intrusion reports reflecting underlying commonalities.

16. The system of claim 14, wherein the integration further comprises invoking countermeasures to a suspected attack.

17. The system of claim 14, wherein the plurality of network monitors include an application programming interface (API) for encapsulation of monitor functions and integration of third-party tools.

18. The system of claim 14, wherein the enterprise network is a TCP/IP network.

19. The system of claim 14, wherein the network monitors are deployed at one or more of the following facilities of the enterprise network: {gateways, routers, proxy servers}.

20. The system of claim 14, wherein the plurality of network monitors includes a plurality of service monitors among multiple domains of the enterprise network.

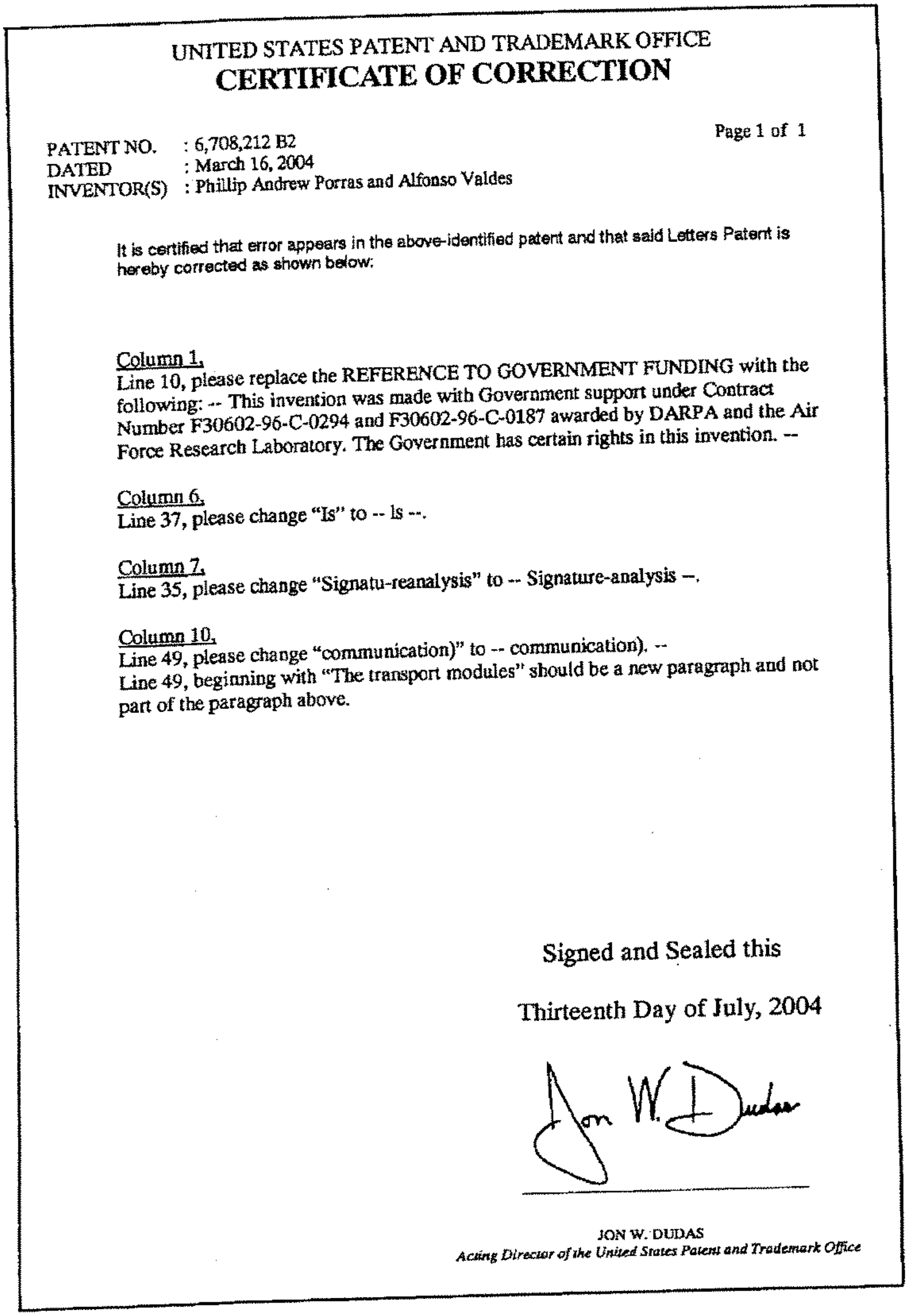
21. The system of claim 20, wherein a domain monitor associated with the plurality of service monitors within the domain monitor's associated network domain is adapted to automatically receive and integrate the reports of suspicious activity.

22. The system of claim 14, wherein the plurality of network monitors include a plurality of domain monitors within the enterprise network, each domain monitor being associated with a corresponding domain of the enterprise network.

23. The system of claim 22, wherein an enterprise monitor associated with a plurality of domain monitors is adapted to automatically receive and integrate the reports of suspicious activity.

24. The system of claim 22, wherein the plurality of domain monitors within the enterprise network interface as a plurality of peer-to-peer relationships with one another.

* * * * *



UNITED STATES PATENT AND TRADEMARK OFFICE

# CERTIFICATE OF CORRECTION

Page 1 of 1

PATENT NO. : 6,708,212 B2
DATED : March 16, 2004
INVENTOR(S) : Phillip Andrew Porras and Alfonso Valdes

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Column 1,
Line 10, please replace the REFERENCE TO GOVERNMENT FUNDING with the following: -- This invention was made with Government support under Contract Number F30602-96-C-0294 and F30602-96-C-0187 awarded by DARPA and the Air Force Research Laboratory. The Government has certain rights in this invention. --

Column 6,
Line 37, please change "Is" to -- is --.

Column 7,
Line 35, please change "Signatu-reanalysis" to -- Signature-analysis --.

Column 10,
Line 49, please change "communication)" to -- communication). --
Line 49, beginning with "The transport modules" should be a new paragraph and not part of the paragraph above.

Signed and Sealed this

Thirteenth Day of July, 2004

Jon W. Dudas

JON W. DUDAS
*Acting Director of the United States Patent and Trademark Office*

SRI International

**Richard H. Abramson**
**Vice President, Legal & Business Affairs**
**General Counsel**

March 31, 2004

**VIA FEDERAL EXPRESS**

Mr. Richard Macchia
Sr. Vice-President and
Chief Financial Officer
Internet Security Systems, Inc.
6303 Barfield Road
Atlanta, Georgia 30328

Re: SRI Network Security Patents

Dear Mr. Macchia:

I am the VP of Legal and Business Affairs and General Counsel of SRI International. For over 50 years, SRI has performed research and development work for thousands of government and commercial clients across a wide variety of fields, including information security. As discussed more fully below, SRI has a number of fundamental patents in the network-based intrusion detection and prevention area, and we would like to explore with you a productive and mutually-beneficial approach to realizing their value.

A little background is probably in order. Over the last decade, SRI has done extensive work for the government in the area of cybersecurity and network-based intrusion detection and prevention. Based upon that work, SRI now has five issued patents and several other pending applications, which cover, among other things, network-based surveillance, hierarchical event monitoring and analysis, and alert detection and management.

SRI's patents in this area have record priority dates going back to 1998. The scope and breadth of their claims vary: some are quite broad, while others are more narrowly focused on particular solutions. Based on our analysis of the prior art, as well as the PTO's approval after review of extensive prior art, we believe our claims to be valid.

SRI believes that a significant number of network-based security products, both software and appliances, infringe multiple claims of SRI's patents. We intend to embark on a major licensing program, focusing on firewall, IDS and integrated network security software and appliances from major vendors in the field.

ISS is one of the companies to whose products -- including without limitation its Proventia™ integrated gateway and network security products and its RealSecure™ network security products-- our patents are relevant. Other ISS products may also be implicated, though we haven't yet done the analysis.

We would like to discuss approaches to this situation that have the potential to be mutually beneficial both to ISS and to SRI. In that spirit, and after you and your counsel have had an opportunity to review the patents and their file histories, we would suggest that a meeting be arranged to discuss this matter face-to-face. We would agree, of course, that anything said by either SRI or ISS at such a meeting would be confidential and inadmissible.

I am enclosing copies of SRI's issued patents in this area. The file histories are easily available from the U.S. Patent Office. If you are unable to find any of the cited prior art, let us know and we would be happy to provide you with a copy.

I look forward to hearing from you, and to discussing this matter with you and your team in greater detail. Please be advised that we have initiated discussions regarding this matter with another company, and that there may be only a narrow window during which arrangements involving any kind of exclusivity may be available. If ISS is interested in engaging in discussions of such a nature, therefore, it would be useful to commence those discussions as promptly as possible.

Very truly yours,

Richard H. Abramson
VP Legal & Business Affairs

RHA/rlr

Enclosure

AO 440 (Rev. 10/93) Summons in a Civil Action

# UNITED STATES DISTRICT COURT

Northern District of Georgia

INTERNET SECURITY SYSTEMS, INC.
Plaintiff,

V.

SRI INTERNATIONAL,
Defendant.

## SUMMONS IN A CIVIL CASE

CASE NUMBER: 1 04 CV 2402

TO: (Name and address of Defendant)

SRI INTERNATIONAL
c/o CT Corporation System
1201 Peachtree Street, N. E.
Atlanta, Georgia 30361

**YOU ARE HEREBY SUMMONED** and required to serve upon PLAINTIFF'S ATTORNEY (name and address)

Holmes J. Hawkins III, Esq.
King & Spalding LLP
191 Peachtree Street, Suite 4900
Atlanta, Georgia 30303-1763

an answer to the complaint which is herewith served upon you, within Twenty (20) days after service of this summons upon you, exclusive of the day of service. If you fail to do so, judgment by default will be taken against you for the relief demanded in the complaint. You must also file your answer with the Clerk of this Court within a reasonable period of time after service.

LUTHER D. THOMAS
CLERK

8.17.04
DATE

(By) DEPUTY CLERK

AO 440 (Rev. 10/93) Summons in a Civil Action

## RETURN OF SERVICE

| | |
|---|---|
| Service of the Summons and complaint was made by me[1] | DATE |
| NAME OF SERVER *(PRINT)* | TITLE |

*Check one box below to indicate appropriate method of service*

G Served personally upon the third-party defendant. Place where served: ______________________

G Left copies thereof at the defendant's dwelling house or usual place of abode with a person of suitable age and discretion then residing therein.

Name of person with whom the summons and complaint were left: ______________________

G Returned unexecuted: ______________________

G Other (specify): ______________________

## STATEMENT OF SERVICE FEES

| TRAVEL | SERVICES | TOTAL |
|---|---|---|
| | | |

## DECLARATION OF SERVER

I declare under penalty of perjury under the laws of the United States of America that the foregoing information contained in the Return of Service and Statement of Service Fees is true and correct.

Executed on ______________________ ______________________

Date Signature of Server

______________________

Address of Server

[1] As to who may serve a summons see Rule 4 of the Federal Rules of Civil Procedure.